May 6–10, 2007

San Jose Convention Center

San Jose, California, USA

Session: J10

DB2 Auditing and the Need for NO Secrecy

IDUG® 2007 North America

Steen Rasmussen
CA

May 09, 2007 01:40 p.m. – 02:40 p.m.

Platform: DB2 for z/OS

GoFurther

Abstract

- Auditing has always been an issue. The past few years have really widened the scope and auditing is now a part of everybody's daily tasks. In the past we only had to worry about what the SYSADM's were doing and which users should be allowed to use which secondary authorization-id's. Dozens of regulatory requirements are now present and it has become mandatory to define processes making sure none of "the rules" are violated.
- This presentation will illustrate how some of the CA DB2 solutions can be used to comply to the regulatory requirements and automate the processes needed to satisfy these requirements. In many cases it is more productive to not implement a lot of restrictions, but instead have solid reporting processes in place.
  The solutions described can be implemented without the CA DB2 products installed and licensed, but more labor can be expected.
- This presentation will use real life scenarios to illustrate different solutions implemented to fulfill the many different requirements IT organizations are facing today. Beside from listing some of the requirements, solutions will be outlined illustrating how an IT organization can satisfy the need to describe how any object was defined at any date in the past, how schema changes can be documented. Also – how basic auditing can be automated to generate reports detailing what certain user types have done when and where, and which commands have been executed as well as who has performed authorization-id switching among other tasks.

Agenda

- Regulatory Compliance issues / challenges
- Tracking and reporting DDL changes
- Table definitions “On Demand”
- Tracking and reporting DML violations / executions
- Challenges retaining the DB2 Log
- Tracking and Reporting what is not in the DB2 Log

Regulatory Compliance Issues

- Protecting assets no matter which Regulatory Act – not only Sarbanes Oxley
  - You're left to comply with myriad information security regulations, including:
    - Nearly three dozen state breach notification laws
    - HIPAA Security Rule mandating the protection of personal healthcare records
    - Gramm-Leach-Bliley Act covering personal financial information
    - PCI Data Security Standard requiring the lockdown of credit card information for merchants
    - Basel II for financial organizations
    - ……..
  - Practically every organization is being held to higher standards of information security.

A couple of years ago everybody were talking about SOX compliance issues. Since then dozens of additional regulatory requirements have popped up dealing with every kind of business from financial companies to personal information related to each individual's information in the health care business. Also – new regulations are in place to secure data can be recovered in time without the loss of integrity and the "business itself".

A few years ago some companies were forced to report what some individual IT employees were doing or who updated certain tables – today's requirements go way beyond these "simple" reporting issues and are now involving every aspect within the IT department. It is just a matter of time until every IT department is facing some kind of audit challenge.

Regulatory Compliance Issues

- This concerns you as a DBA or network administrator responsible for database security because the database is where most, if not all, sensitive information covered by these laws and regulations is stored. The database is where the "gold" is, and it's also where you can focus your attention on making improvements.
- Not only a matter of creating Audit Trail Reports
  - Financial Data
  - Risk Management
  - Personal Information
- Regulatory issues can be broiled down to four basic issues – all giving the IT departments headaches / challenges
  - Protection of assets (the table content)
  - Reporting of “who”, “what”, Where”, “how”,……
  - Retention of data, logs, audit trails,……..
  - Scrambling of data which can be considered part of the first challenge, but requires a complete different set of techniques

5

So why do we DBA’s care – we’re not responsible for the data and the application code ? Well – all this sensitive data resides in a nice database named DB2 – and we are the people maintaining the database, implementing objects and perform schema management – and we have a lot of power in terms of authorizations. We are the people who will be forced to assist implementing the required processes and procedures to make sure the business stays compliant.

The increased regulatory requirements do not only deal with changes to sensitive data, but the challenges we’re facing can be divided into four groups where we need to have the appropriate processes in place:

1) Protection of data
2) Reporting who touched the sensitive data and what was changed
3) Retaining DB2 logs, audit trail reports and historical data
4) Data encryption to scramble sensitive data

Let’s have a closer look at these different topics to see what we as DBA’s can do to stay compliant.

Protecting Assets - Challenges

- What are we doing to protect the assets
  - Install/implement security procedures
    - External DB2 procedures like RACF, CA-ACF2, CA-Topsecret etc.
    - Internal DB2 procedures like native DB2 security, MLS (Multi Level Security in DB2 V8) and ROLES in DB2 9
- What is the challenge seen from the IT Department perspective?
    - The DBA and Systems Programmers job is becoming difficult
    - "Correcting" application failures by updating data using ad-hoc SQL queries (SYSADM issues)
    - Utilities need to be executed – Unload, Recovery, Load
    - Objects need to be created, Dropped, Altered,…….
    - Not to mention the IT department also will become responsible for what everybody else "is doing" (prohibit, protect and specifically reporting to make sure nothing "illegal" is happening)

The first topic is how to protect the assets – also known as DB2 table data. In the early days of DB2 only DB2 native security could be used via GRANT. Later – external security was added making it possible to control access using external security packages like CA-ACF2, CA-Topsecret and IBM RACF, making it possible to use masking to ease the security administration.

DB2 V8 introduced MLS (Multi level Security) so it is possible to protect certain rows based on a hierarchy of security labels. However – even though this feature makes sure only the authorized users can read the data they are authorized to read, it introduces some new challenges when dealing with utilities like REORG, UNLOAD etc.

DB2 9 will provide further security features by introducing ROLES which especially will help in the day-to-day DB2 tasks.

This being said – as a DBA we need to reorg, unload, load, recover data – and this is data we really are not allowed to touch. Also – often it is necessary to "manipulate" data due to logical application errors, and often the only people with the appropriate authorization and tools are the DBA's (another compliance issue).

So since the DBA is the person who knows DB2, has access to the tools and utilities – the DBA either already is or will be responsible to implement a lot of the processes needed to be or stay compliant.

Protecting Assets - Challenges

- We can implement “road blocks” to prevent unauthorized access
  - Grant’s
  - Secondary auth-id’s (but then we need to track WHO changes auth-id)
  - DDCS – who can execute DDL like Alter, Create and Drop using which Plans / Packages
- But it’s only a “snap shot” of permissions
  - We can change it back once the auditors are out the door

- We need to track WHO changed what at any point in time
- We need to track WHO created, altered, dropped objects
- Retaining the DB2 log for ten years or more is in most cases not feasible

Sure we can make certain only the “correct people” have access to the data, but when secondary ID’s are used, we then need to track who used what ID and what was changed. BUT – remember one thing – the authorizations present is only a snapshot and does not reflect what was the truth one minute ago or what will be the truth tomorrow.

The same goes for schema management – the demand to track what was changed, created and dropped is increasing, and DB2 does not provide appropriate control mechanism using DDCS.

So despite the DB2 engines capabilities, there is indeed a need to track which primary authorization-id’s changed what data, and who changed which objects from what to what. Please keep one issue in mind – DB2 does record (in the catalog) when an object was created, when an object was changed last time – but dropped objects are not recorded and you only see the latest ALTER to an object..

Another issue – since it might be necessary to be able to report on data changes or DDL changes five, ten or fifteen years ago – keeping the DB2 log for such a long time is in most cases a major issue, and often not possible at all.

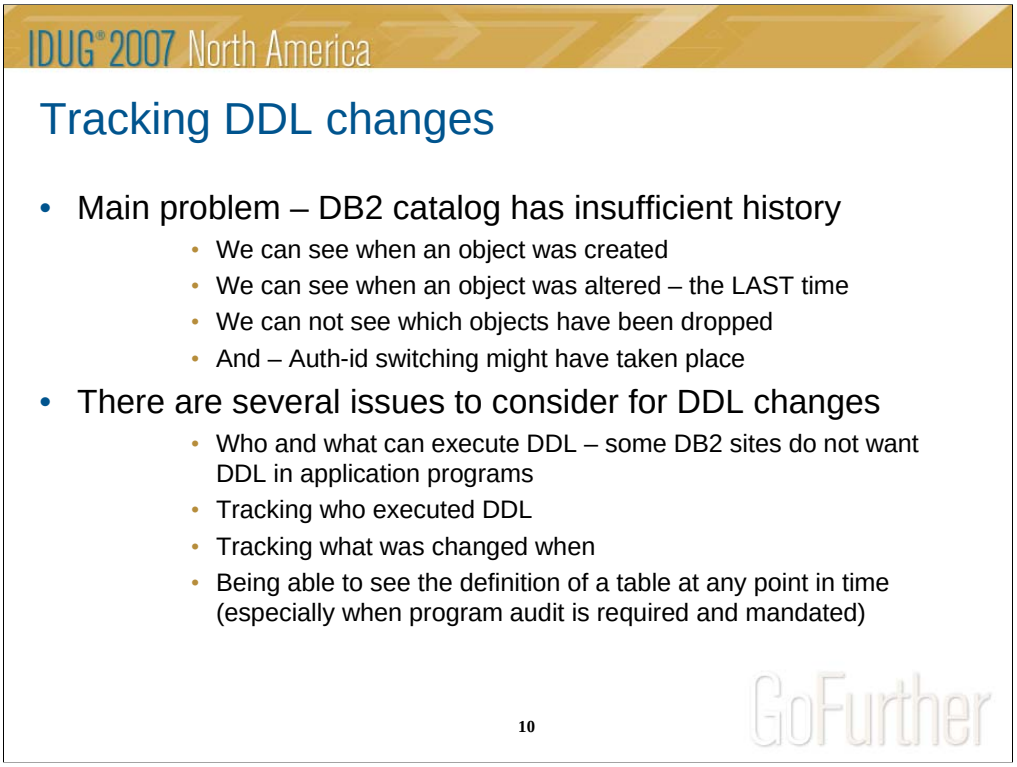
Solutions to stay compliant

- Let's look at some solutions to solve these issues and still be in a position where "we can get the job done"
- The keywords are Preventing, Tracking and Reporting
  - DDL changes
  - DML executions and SQL in general
  - Retaining the DB2 Log
  - Scrambling/encrypting data for testing
  - Anything else . . . . . .

From a DBA perspective, the following disciplines can be used to stay compliant and to satisfy most of the regulatory requirements:

1) Track which objects were changed, what was changed and what was dropped
2) Track who changed what when it comes to "sensitive data"
3) Some methods to retain the essential parts of the DB2 log in order to report on anything back in time
4) Encryption / scrambling of data – and when it's necessary
5) Other issues to consider in order to satisfy auditing requirements, which can be DB2 commands, SQL-ID switching, Utilities executed, access denied etc.

Let’s first look at Schema Changes and what can be done to stay compliant.



One area which has not had a lot of attention over the past two decades - but the past couple of years have increased the requirement to report and track – is object changes, and the problem somehow is related to the DB2 catalog itself:

1) How to control and report on who can execute which DDL
2) The DB2 catalog only shows when an object was created or altered the last time – but not necessarily what was changed
3) Dropped objects do not appear in the DB2 catalog
4) There is an increasing need / demand to illustrate the exact definition of a table at any point in time in order to map the object definition to the application. Beside from the auditing perspective – this could be beneficial too if there is a need to restore a table to a prior point in time to "recreate a row image"

Where can DDL be executed

- DB2 offers Data Definition Control Support (DDCS)
  - Enabled via DSNTIPZ panel (DB2 Admin Guide chapter 10)
  - Two tables created to control DDL execution
    - DSN_REGISTER_APPL is used to specify which Plans and Packages can execute DDL for some object types
    - DSN_REGISTER_OBJT is used to specify which object types and optional object names can be created, altered, dropped by which plans/packages
    - Wildcarding is supported

Object	CREATE statement	ALTER statement	DROP statement
Alias	CREATE ALIAS		DROP ALIAS
Database	CREATE DATABASE	ALTER DATABASE	DROP DATABASE
Index	CREATE INDEX	ALTER INDEX	DROP INDEX
Storage group	CREATE STOGROUP	ALTER STOGROUP	DROP STOGROUP
Synonym	CREATE SYNONYM		DROP SYNONYM
Table	CREATE TABLE	ALTER TABLE	DROP TABLE
Table space	CREATE TABLESPACE	ALTER TABLESPACE	DROP TABLESPACE
View	CREATE VIEW		DROP VIEW

Snapshot from IBM DB2 Admin Guide - (Note not every object type is Supported)

DB2 has since the early days provided a feature to assist in controlling which applications and users are allowed to execute DDL statements. This feature is named DDCS – or – Data Definition Control Support which can be enabled using the appropriate DSNZPARM parameters. The problem is not every object type is supported to be controlled.

The good new is – this feature is really good at controlling which applications (packages and plans) are allowed to execute DDL statements against which objects.

Exploiting DDCS has some shortcomings, so it might still be necessary to track the exact changes (done when and by whom) for auditing purposes (more about this later).

Tracking who executed DDL

- Every DDL command results in Insert, Delete, Update of the catalog tables – and logged like any other DML statements
- Unicenter Log Analyzer offers several different methods to report DDL changes
- Resume-processing can be used to automate this process
  (will be covered in greater detail later)

```
LARDDL R11.5  ----------- DDL Activity Report Options ---------- 07/01/05 16:53

FORMAT OPTIONS
Output Format   ==> r  ( R - Rept, D - Redo DDL, U - Undo DDL)
Level of Detail ==> D  ( S - Smry, D - Detail, T - Tot, I - ImageCopy)
Customize Rept  ==> N  ( Y , N , U )  Form ==> ________  Creator ==> ________

LOG DATA FILTER OPTIONS ( I - Include Data Filters, X - Exclude Data Filters )
Plan Filters    ==> _      Connection Ids   ==> _     URID filters   ==> _
Primary Authids ==> _      Correlation Ids  ==> _
Join Operator   ==> AND ( AND / OR )

MISCELLANEOUS OPTIONS
Grant/Revoke    ==> X ( I , X )       Generate Table OBIDs  ==> Y ( Y , N )
Bind /Rebind    ==> X ( I , X )
```

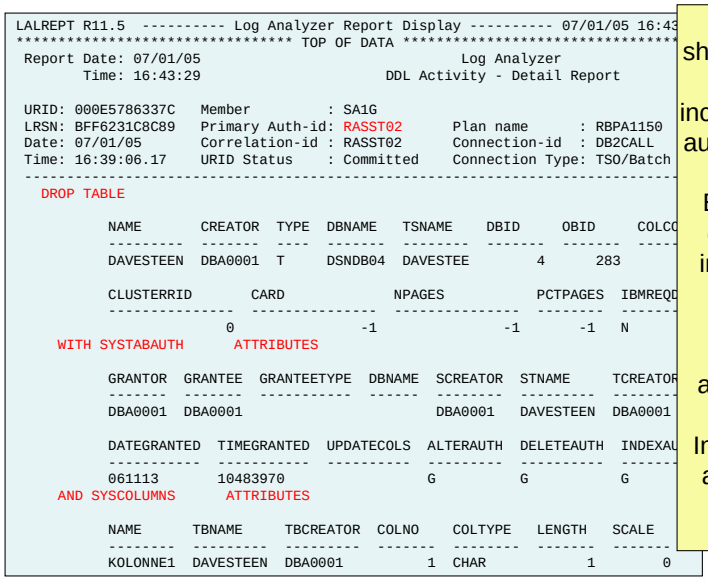

When CREATE, ALTER and DROP statements are executed, these will cause INSERT, DELETE and UPDATE statements against the catalog, and by then be in the DB2 log.

Unicenter Log Analyzer can generate summary or detailed reports describing exactly which objects have been created, dropped or altered and by whom.

Instead of submitting these kind of reports every hour or every day, in order to automate this process so no intervention is needed, a strategy can be defined using RESUME processing, so any execution will start where the previous ended. This will be covered in more detail later.

First – let us have a look at on of the generated reports

Tracking who executed DDL



```
LALREPT R11.5  ---------- Log Analyzer Report Display ---------- 07/01/05 16:43
******************************** TOP OF DATA ********************************
 Report Date: 07/01/05                          Log Analyzer
       Time: 16:43:29                    DDL Activity - Detail Report

 URID: 000E5786337C   Member         : SA1G
 LRSN: BFF6231C8C89   Primary Auth-id: RASST02      Plan name      : RBPA1150
 Date: 07/01/05       Correlation-id : RASST02      Connection-id  : DB2CALL
 Time: 16:39:06.17    URID Status    : Committed   Connection Type: TSO/Batch
 ------------------------------------------------------------------------------
   DROP TABLE

          NAME       CREATOR  TYPE  DBNAME   TSNAME    DBID     OBID     COLCO
          ---------  -------  ----  -------  --------  -------  -------  -----
          DAVESTEEN  DBA0001  T     DSNDB04  DAVESTEE        4      283

          CLUSTERRID       CARD             NPAGES           PCTPAGES  IBMREQD
          ---------------  ---------------  ---------------  --------  -------
                         0               -1               -1        -1  N
     WITH SYSTABAUTH      ATTRIBUTES

          GRANTOR  GRANTEE  GRANTEETYPE  DBNAME  SCREATOR  STNAME     TCREATOR
          -------  -------  -----------  ------  --------  ---------  --------
          DBA0001  DBA0001                       DBA0001   DAVESTEEN  DBA0001

          DATEGRANTED  TIMEGRANTED  UPDATECOLS  ALTERAUTH  DELETEAUTH  INDEXAU
          -----------  -----------  ----------  ---------  ----------  -------
          061113       10483970                G          G           G
     AND SYSCOLUMNS      ATTRIBUTES

          NAME      TBNAME     TBCREATOR  COLNO    COLTYPE   LENGTH   SCALE
          --------  ---------  ---------  -------  --------  -------  -------
          KOLONNE1  DAVESTEEN  DBA0001          1  CHAR            1        0
```

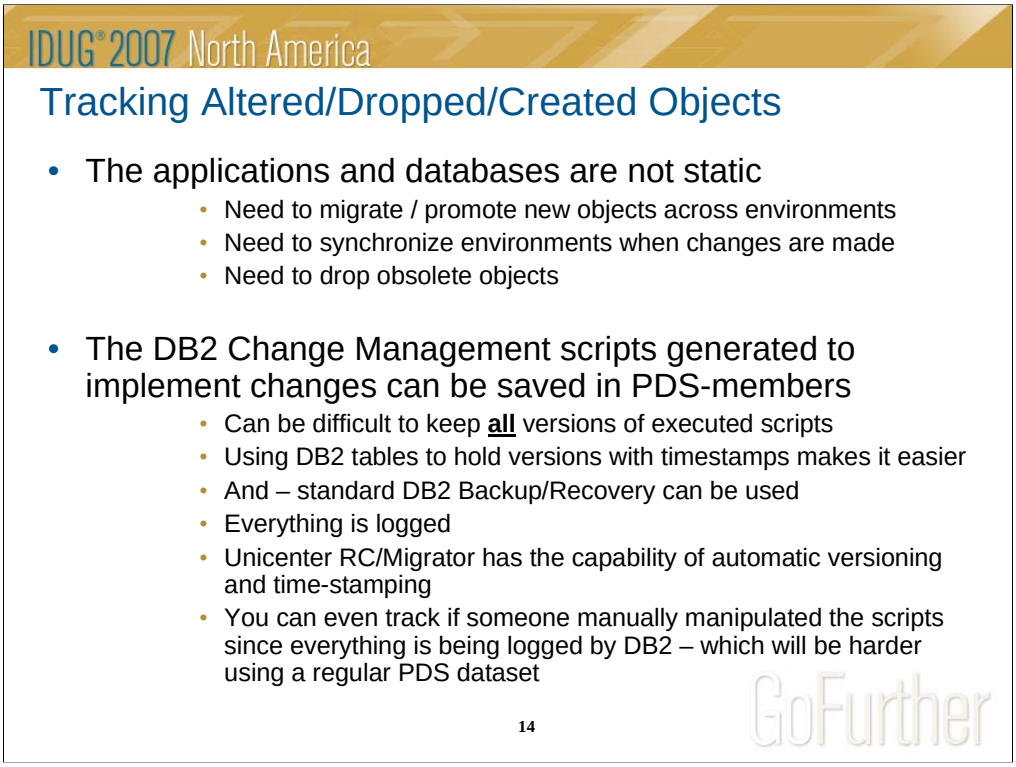
This partial report shows part of a DROP table execution – including the PRIMARY auth-id, date, time etc..

Every row updated, deleted or inserted in the catalog tables are reported – like any regular DML statement from an application program.

Instead of this report, a PDS member can be generated with DDL statements.

The detail report illustrates when the object change happened, who executed the statement (both primary and secondary id) as well as the content of every catalog column for the object.

An alternative to have this detailed report generated, is to have Log Analyzer generate the actual DDL statement, and place it in the report or as a PDS-member.



Most application databases are not static – meaning changes are implemented over time in order to change columns, create objects, drop objects etc.

There is an increased demand to illustrate which changes have been implemented to which objects. Even though solid schema management procedures are implemented and enforced, it can be a challenge to keep every change (schema management script) in PDS members and it can be difficult to find the changes when required due to auditing. Instead of PDS members, it can be beneficial to save everything in a DB2 table. The advantage is normal DB2 image copy procedures can be used to make sure everything is retained. Using Unicenter RC/Migrator to manage the schema changes, Managed Output can be used to store every script in a DB2 table.

Another advantage of using a DB2 table as the storage is, if anyone is trying to change the scripts outside the implemented schema management procedures, these changes (as well as the user-id) is logged by DB2 and the previous mentioned Unicenter Log Analyzer strategy can be implemented to report on these violations.

Tracking Altered/Dropped/Created Objects

- Managed output is an option within Unicenter RC/Migrator
- Every generated script is saved in DB2

```
RMS1 R11.5  -------------- RC/M Strategy Services -------------- 07/01/12 12:45
COMMAND ===>                                                   SCROLL ===> CSR

DB2 SSID ===> D81A
STRATEGY ===> *        CREATOR ===> RASST02  TYPE ===> *    SRC SSID ===> *
------------------------------------------------------------------------ RASST02
                                             T S SRC  +---- LAST UPDATE ----+
O STRATEGY DESCRIPTION              CREATOR  P O SSID   USER     DATE   TIME
_ KMDCONV  DB=Sales synchronization  RASST02  C U D81A RASST02  05/12/22 06:50
  _ * MANAGED OUTPUT *
     COMMENT: Change management issue #12443
  _ * MANAGED OUTPUT *                                          RASST02  06/02/12 07:13
     COMMENT: Change management issue #15122
  _ * MANAGED OUTPUT *                                        RASST02  06/03/05 07:19
     COMMENT: Change management issue #15770
  _ * MANAGED OUTPUT *                                        RASST02  06/06/09 17:23
     COMMENT: Prepare application rollout for #18905
_ TCPMIG1  Migrate pay appl          RASST02  M U D81A RASST02  05/11/06 07:29
  _ PTIDEVL.VIRTUEL.DB2(TCPMIG1)                              RASST02  05/12/18 07:32
  _ PTIDEVL.VIRTUEL.DB2(TCPMIG1)                              RASST02  06/10/17 07:50
  _ PTIDEVL.VIRTUEL.DB2(TCPMIG1)                              RASST02  06/11/21 09:20
  _ PTIDEVL.VIRTUEL.DB2(TCPMIG1)                              RASST02  06/12/02 09:26
******************************* BOTTOM OF DATA ********************************
```

Browse the script / analysis report next page

The strategy named TCPMIG1 has been analyzed (script generated) four times, but all generated scripts are stored in the same PDS-member in dataset ptidevl.virtuel.db2, and there is no history of the previous generated scripts – unless a new member is specified prior to generating the script.

The other strategy (KMDCONV) is using Managed Output, where the generated script (worklist) is stored in a DB2 table, so every time the script is generated, a new entry is saved in the managed output table and can be browsed/edited at any point-in-time.

The next slide will illustrate the content of one of the generated scripts/worklists from the managed output table.

Tracking altered/dropped/created Objects

- The Report Analysis option also documents changes so it's not necessary to review DDL

```
 BROWSE -- KMDCONV~RASST02~RASST02~20051222~06515150  Line 00000000 Col 001 080
 Command ===>
******************************** Top of Data ***************
.CONTROL SN(RASST02,KMDCONV)

--============================================================================
-- VRCM1 SP1    CA-DB2   RC/COMPARE    ANALYSIS REPORT      05/12/22 06:51
--============================================================================
--STRATEGY INFORMATION:
--STRATEGY ==> KMDCONV   DESCRIPTION  ===> DB=Sales synchronization
--  Table RASST02.CONV1 Changes:
--
--  ------------ Table Column Changes ------------------
--  Field       Currently                    Changed To
--  -----       ---------                    ----------
--  NAME        CARD                         (NOT CHANGED)
--  TYPENAME    INTEGER                      CHAR
--
--
--                           TOTAL CHANGED OBJECTS:         1
```

Instead of browsing a PDS member, this is the script read from the DB2 table where key is strategy/worklist and a timestamp.

The analysis Report shows every attribute changed – what it was and what it was changed to.

Browsing one of the generated scripts/worklists, you can see the browsed script is timestamped from the actual analysis time.

Prior to the actual statements in the script, the analysis report is generated illustrating exactly which schema changes are contained in this script. In this case, we can see table=RASST02.CONV1 has a column-name=CARD which is not changed but the attribute has been changed from INTEGER to CHAR. Every change is described in the header, meaning it is not necessary to be a DB2 expert and browse through the entire script to see what was changed. This is another good point when dealing with auditors.

May 6–10, 2007

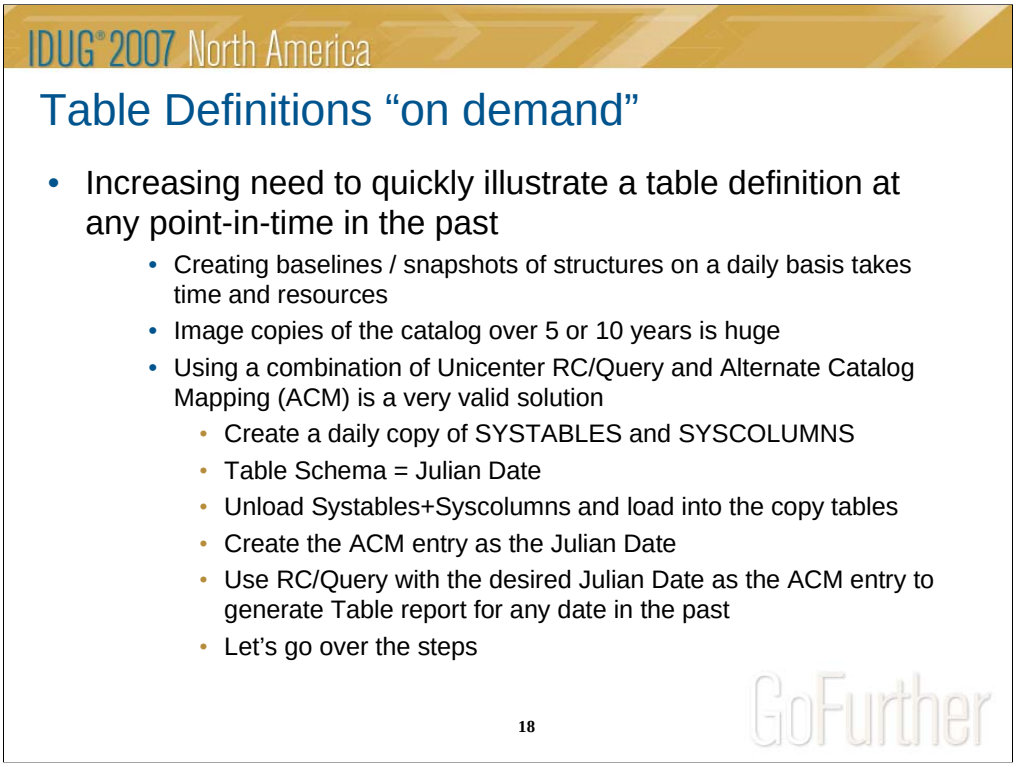
San Jose Convention Center

San Jose, California, USA

Table Definitions “on demand”

IDUG® 2007 North America

GoFurther



Many DB2 sites have the need to be able to illustrate a table definition at any point-in-time in the past – not only because of auditing requirements. Another reason is to map the actual application programs with the physical database structure.

One method is to create a baseline for every database every day, but it’s a costly process.

Another method could be to keep image copies of the catalog over the required time, which also can be a costly method.

Part of the Value Pack component is ACM (Alternate Catalog Mapping) which offers the ability to entirely or partly us non DB2 catalog tables when reports are generated. This feature can be used to create a “daily view of table definitions“ and still us the online/batch reporting facility.

1) Every day a new SYSTABLES and SYSCOLUMNS is created

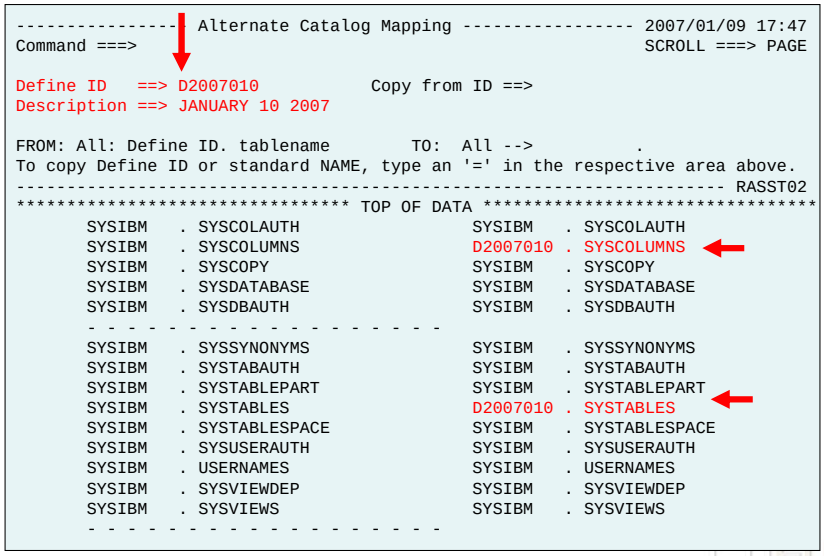
2) Use table creator=Julian Date (ease of use)

3) Insert into NEW tables SELECT * from SYSIBM tables

4) Step into the Unicenter DB2 products Main Menu where option M is used to create a new ACM entry=JULIAN DATE

5) Update this ACM entry to point at these new tables for SYSTABLES and SYSCOLUMNS while and SYSIBM for the rest.

Table Definitions "on demand" – Create ACM entry

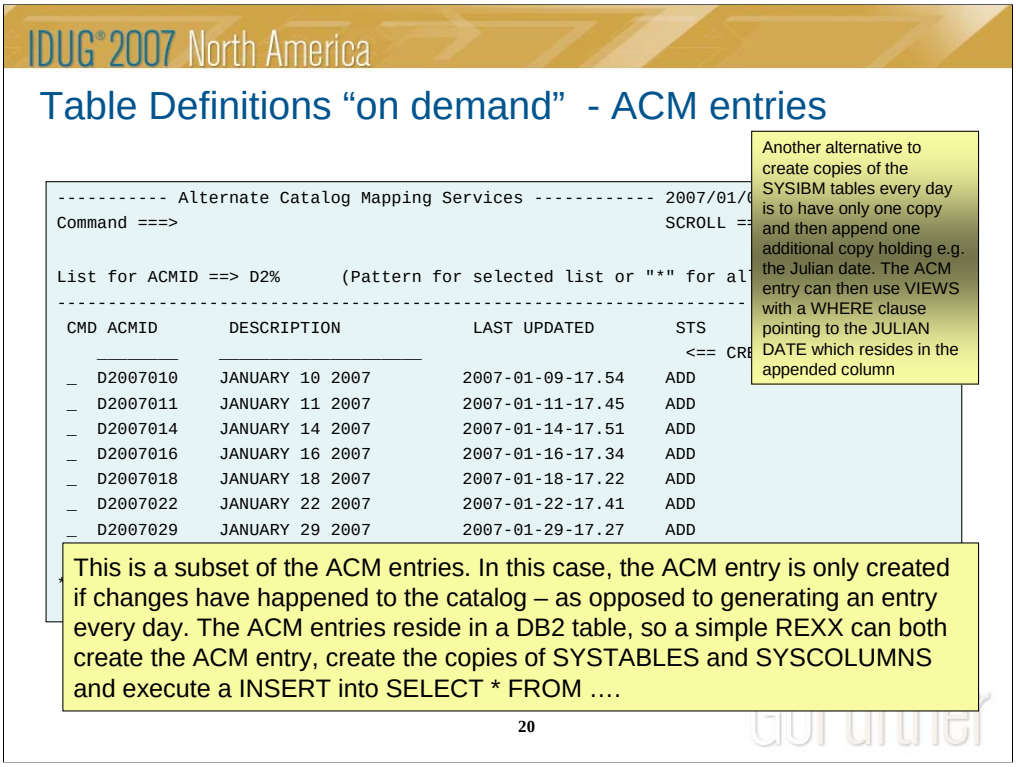


This screen shot illustrates an ACM entry=D2007010 (Julian date for January 10 2007) has been created. Any ID can be used, but I prefer to use something simple to use, so YYMMDD could be another alternative.

The DB2 catalog tables are listed on the left hand side, while the MAPPED tables (could be views as well) are listed on the right hand side.

Please not only SYSCOLUMNS and SYSTABLES have been mapped, so when this ACM-entry is used, the underlying SQL statements will select from D2007010.SYSTABLES and D2007010.SYSCOLUMNS while all other tables will be the original DB2 catalog tables.

In this case, these two D2007010 tables will have to be created and populated with the DB2 catalog content for this specific date.



This screen shot illustrates the different ACM-ID’s created over time. In this case the ACM-ID is only created IF any changes has happened to tables and/or columns. For the ease of use – it could be beneficial to ALWAYS create the ACM entry for the cost of DASD and number of objects.

The ACM-ID’s exist in a DB2 table, so it is pretty easy to create these automatically. All that is needed is a couple of INSERT statements into the underlying tables and the INSERT into the ACM tables by selecting everything from SYSCOLUMNS and SYSTABLES.

An alternative to create new tables every day could be to us a table controlled partitioned table where the limitkey is e.g.Julian Date. Then the ACM entry should reference two views instead and then use a where clause to only reference the rows for the specific day. This method will eliminate two tables every day, and the partitioned table can be ROLLED to clean up obsolete / old entries.

Table Definitions “on demand” Using the ACM entry

- Let us assume we need to look at how a table was defined January 19 2007
  - You can calculate the Julian Date or ..
  - You can browse the ACM entries to find the desired ACM-ID prior to entering Unicenter RC/Query

```
----  r11.5  ------- Unicenter DB2 Products Main Menu  ------ 2007/01/09 18:08
OPTION ===> 1                                              SCROLL ===> PAGE

DB2 SSID ==> D81A LOCATION ==> LOCAL                     DB2 VERSION : V8R1M
ACM      ==> yes  ACMID    ==> d2007019 SQLID ==> RASST02

<-> Backup and Recovery                 <-> Report Facility
    _ LA Log Analyzer                       _ R  Report Facility Menu
    _ MM Merge/Modify
    _ Z  Recovery Analyzer              <-> Utilities
                                            _ U  DB2 Object Manager
<-> Database Administration
    _ PX Partition Expert               <-> Value Pack
    _ 1  RC/Query                           _ B  Batch Processor
    _ 2  RC/Migrator                        _ C  DB2 Command Processor
    _ 3  RC/Update                          _ I  Interactive SQL
    _ 4  RC/Secure                          _ M  Alt. Catalog Mapping
    _ 5  RC/Extract                         _ TT Thread Term/Dynam DSNZPARM
    _ 7  Endevor for DB2                    _ Y  Utility Manager
```

This is how in works in praxis. Let us assume we have a need to see a tables definition January 19 2007. This example is using Unicenter RC/Query, but it could be done using a regular SQL SELECT statement, but then you will have to remember which tables to select from and join. ACM=YES is specified and the ACM-ID for the desired date (Julian Date D2007019).

Table Definitions “on demand” Using the ACM entry

- The table definition for the specific date is generated – very simple – only by specifying the date of interest

```
RQTC   R11.5  ---------- RC/Q Table Column Inquiry ---------- 2007/01/09 18:15
COMMAND ===>                                                  SCROLL ===> CSR

 DB2 Object ===> T                     Option  ===> C    Where => N
 Table Name ===> resource            >  Creator ===> *                      >
  Qualifier ===> *                   >  Grantor ===> *                      >
Loc: LOCAL ---------- SSID: D81A ----------RASST02 -        LINE 01 OF 14    >
CMD      TABLE NAM CREATOR   COLUMN NAME         COLTYPE    LENGTH    SCALE  N
________ RESOURCE  NMJAVAQA
________                     RESOURCE            INTEGER         4        0  N
________                     RCLASS              CHAR           16        0  N
________                     RDESCRIPTION        CHAR           48        0  Y
________                     RPERFORMANCEAGENT   INTEGER         4        0  N
________                     RLEVEL0             CHAR           16        0  Y
________                     RLEVEL1             CHAR           16        0  Y
________                     RLEVEL2             CHAR           16        0  Y
________                     RLEVEL3             CHAR           16        0  Y
________                     RIDENTIFIER         CHAR           48        0  N
________                     RADDRESS            CHAR           16        0  Y
________                     RSUBELEMENT         CHAR           32        0  Y
________                     RRELATED1           CHAR           16        0  Y
________                     RRELATED2           CHAR           16        0  Y
****************************** BOTTOM OF DATA ******************************

Options: D=Detail,L=List,P=Plan,S=Synonym,I=Index,C=Column,V=View,MQ=MQT
         O=Obj.Dependency,UA=User Auth,PA=Plan Auth,KA=Pack Auth,PK=Package
         UC=Unique Constraint, DI=Drop Impact, A=Alias, LR=LobRel, TG=Trigger
```

The SQL used to create the report, is using the tables (or views) described inside the ACM entry.

In this case – SQL statements against SYSTABLES and SYSCOLUMNS will be qualified with D2007029. All other tables will be SYSIBM.

The table name in question is entered, which in this case is all tables with the name=RESOURCE.

Since ACM-ID=D2007019 was specified, the SQL statement executed to generate the table definition is joining D2007019.SYSTABLES with D2007019.SYSCOLUMNS – and any other table needed to generate this report is the original SYSIBM tables.

Being able to generate a table definition “on demand” for a specific date in the past will - beside the ability to provide an audit for an application program, also provide the necessary information for restoring previous image copies to an alternate table without having to print the image copy pages to find the internal ID’s (OBID, PSID, DBID).

May 6–10, 2007

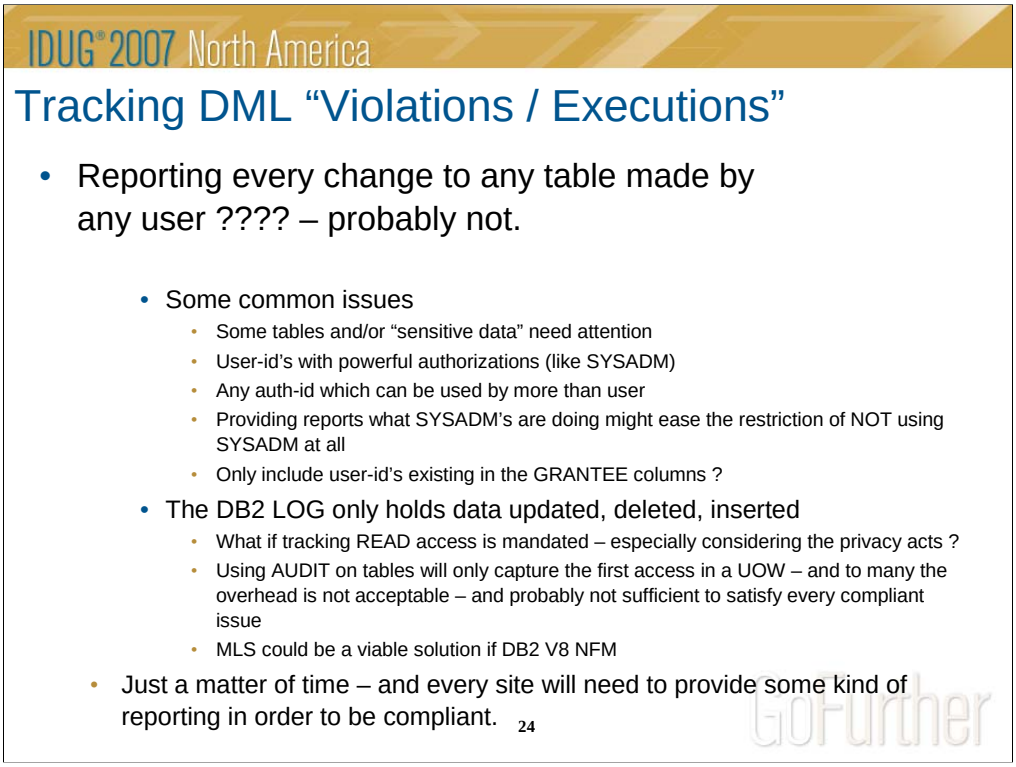
San Jose Convention Center

San Jose, California, USA

Tracking DML
"Violations / Executions"

IDUG® 2007
North America

GoFurther



Being able to report on Who changed What and Where is probably one of the oldest requirements when talking about auditing – and this topic has been important even before all the new regulatory requirements came into play. It has just become even more important and is now an issue which almost every DB2 site needs to deal with. So what is it we need to report on:

1)It's probably not necessary to report on every change for every table

2)In most cases, it's sufficient reporting on tables with sensitive data

3)It's probably not necessary to report what application programs are manipulating due to application coded SQL "fixed" statements

4)SYSADM users are definitely a topic which will have to be used in the reporting. This will often ease the restriction of NOT allowing SYSADM users, and by then make the DBA like easier.

5)We can also limit the reporting to user-id's residing in the GRANTEE column of SYSTABAUTH etc.

The DB2 log holds all DML statements (until DB2 9 comes along), so we can traverse the LOG to report on "what is needed/mandated".

One challenge is SELECT's are not reported in the log, so AUDIT might be needed for some tables – or MLS can be implemented, which will make utilities a challenge.

Tracking DML "Violations / Executions"

- In most cases it's sufficient to report what was changed by whom "outside the business applications".
  - Include/Exclude Plans, Tables, Databases to limit reporting
  - Implement the report types needed to satisfy the auditing requests

```
LARDML R11.5  ----------- DML Activity Report Options ---------- 07/01/12 17:14

FORMAT OPTIONS
Output Format   ==> r  ( R -Rept, S -RedoSQL, U -UndoSQL, L -Load Fmt, P -Updt)
Level of Detail ==> d  ( S - Smry, D - Detail, T - Tot, K - Key, I - ImageCopy)
Order Output By ==> U  ( U , A , P , T , K , KU , RE , UN )
Customize Rept  ==> N  ( Y , N , U )  Form ==> ________  Creator ==> ________

LOG DATA FILTER OPTIONS ( I - Include Data Filters, X - Exclude Data Filters )
Table Filters   ==> i      Database Filters ==> _     Statement Type ==> _
Plan Filters    ==> x      Connection Ids   ==> _     URID filters   ==> _
Primary Authids ==> _      Correlation Ids  ==> _
Join Operator   ==> AND ( AND / OR )

MISCELLANEOUS OPTIONS
Rollbacks         ==> I ( I , X , O )   Specify KeyCols  ==> N ( Y , N )
Catalog Updates   ==> X ( I , X )       Set LOAD Options ==> N ( Y , N )
Report Discards   ==> N ( Y , N )       Discard Limit    ==> 0
Subsequent Opts   ==> N ( Y , N )       Related Updates  ==> N ( Y , N )
Directory Updates ==> X ( I , X )
```

25

Unicenter Log Analyzer can report on any DML statement in the log including LOG YES utilities. In order to limit what is reported, a number of filters can be used to satisfy the auditing needs/demands.

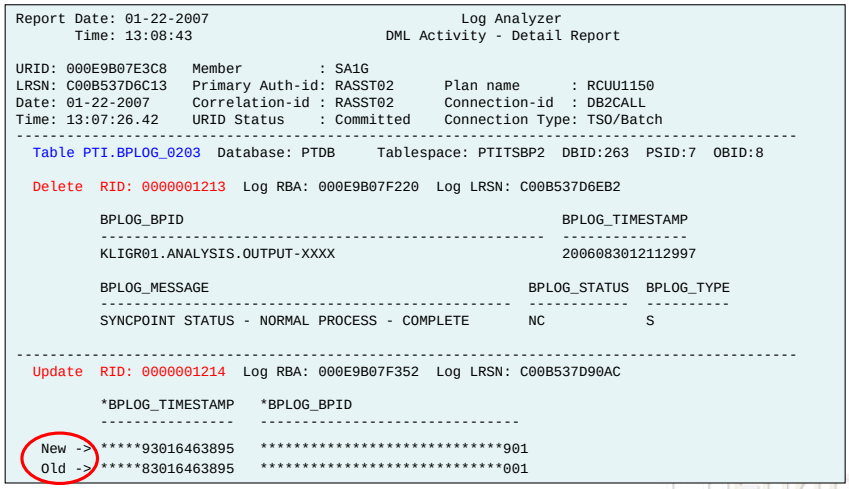
In this case a Detail Report is requested where certain tables will be Included and certain Plans will be eXcluded. Catalog updates are excluded and Rollbacks Included in order to also report on attempted updates/deletes and inserts.

Per default both deletes, inserts and updates are reported. If needed, the report can be generated to only hold DELETEs and UPDATEs or any combination of DML statement types.

Instead of just generating a report, the alternative is to re-generate the actual executed DML statements or creating a "load format file" which then can be loaded into "auditing tables", "history tables" etc.

Tracking DML "Violations / Executions"

- Reports can be requested to illustrate what changed – or the reports can be detailed to see entire before/after image



```
Report Date: 01-22-2007                          Log Analyzer
       Time: 13:08:43                  DML Activity - Detail Report

URID: 000E9B07E3C8   Member         : SA1G
LRSN: C00B537D6C13   Primary Auth-id: RASST02      Plan name      : RCUU1150
Date: 01-22-2007     Correlation-id : RASST02      Connection-id  : DB2CALL
Time: 13:07:26.42    URID Status    : Committed    Connection Type: TSO/Batch
-------------------------------------------------------------------------------------------
  Table PTI.BPLOG_0203  Database: PTDB     Tablespace: PTITSBP2  DBID:263  PSID:7  OBID:8

  Delete  RID: 0000001213  Log RBA: 000E9B07F220  Log LRSN: C00B537D6EB2

          BPLOG_BPID                                          BPLOG_TIMESTAMP
          --------------------------------------------------  ---------------
          KLIGR01.ANALYSIS.OUTPUT-XXXX                        2006083012112997

          BPLOG_MESSAGE                                     BPLOG_STATUS  BPLOG_TYPE
          ------------------------------------------------  ------------  ----------
          SYNCPOINT STATUS - NORMAL PROCESS - COMPLETE      NC            S

-------------------------------------------------------------------------------------------
  Update  RID: 0000001214  Log RBA: 000E9B07F352  Log LRSN: C00B537D90AC

          *BPLOG_TIMESTAMP   *BPLOG_BPID
          ----------------   ------------------------------

   New -> *****93016463895   ****************************901
   Old -> *****83016463895   ****************************001
```


In the case a Detailed Report was selected, so the report generated holds all DML statements executed by the USER-ID's included for the tables included where the PLAN is not in the exclude list.

For table PTI.BPLOG_0203, user=RASST02 (which also is the primary auth-id) has inserted a row, and the entire row image is included.

The same user has also updated a row on the same table. This update is reported as partial, where only the changed bytes and changed columns are reported (due to the table not defined with Data Capture Changes and Image Copy Detail reporting was not used).

If the table was defined with Data Capture Changes, the DB2 log would have held the entire before and after image. If the report had been requested using Image Copy Detail (and Data Capture Changes were not active), Unicenter Log Analyzer would also have listed the entire before and after image by reading the most recent image copy and applying log-records to the row from the image copy point-in-time up to the update RBA.

Tracking DML “Violations / Executions”

- The previous example illustrated the detailed report and what was changed, while this report generates REDO SQL where the entire before/after image is listed

```
EDIT       RASST02.IDUG10.REDOREP                          Columns 00001 00072
------------------------------------------------------------------------------
000138  INSERT INTO PTI.PTLOG_MAIN_0102
000139       ( LOG_USER , LOG_TIMESTAMP , LOG_ID , LOG_CREATOR ,
000140         LOG_OBJECT , LOG_TYPE , LOG_FUNCTION , LOG_OBJECT_CREATOR ,
000141         LOG_OBJECT_NAME , LOG_OBJECT_PART )
000142    VALUES
000143       ( 'RASST02 ' , '2007012213054002' , 'RCUPDATE' , '        ' ,
000144         '                               ' , 'TABLE       ' ,
000145         'EDIT        ' , 'PTI' , 'BPLOG_0203' , 0 )        ;
000147
000148     DELETE FROM PTI.BPLOG_0203
000149    WHERE BPLOG_BPID =
000150          'KLIGR01.ANALYSIS.OUTPUT-XXXX                          '
000151      AND BPLOG_TIMESTAMP = '2006083012112997'
000152      AND BPLOG_SYSID = 'SA1G'
000153      AND BPLOG_USERID = 'KLIGR01 '
000154      AND BPLOG_MESSAGE =
000155          'SYNCPOINT STATUS - NORMAL PROCESS - COMPLETE     '
000156      AND BPLOG_STATUS = 'NC'
000157      AND BPLOG_TYPE = 'S'
000158      AND BPLOG_SYNCID = 0
000159      AND BPLOG_ST_CREATOR = '        '
000160      AND BPLOG_ST_NAME = '        '
000161      AND BPLOG_UTILITY_ID = '                '
000162      AND BPLOG_UTILITY_SSID = '    '   ;
```

27

The previous slide illustrated the detailed report and who changed which tables/columns, when the changed happened and via which plan. As mentioned earlier, it is possible to actual re-generate the executed statements, and this example has been generated using REDO SQL.

For auditing purposes the previous report is the appropriate reporting mechanism, since the following information is available:

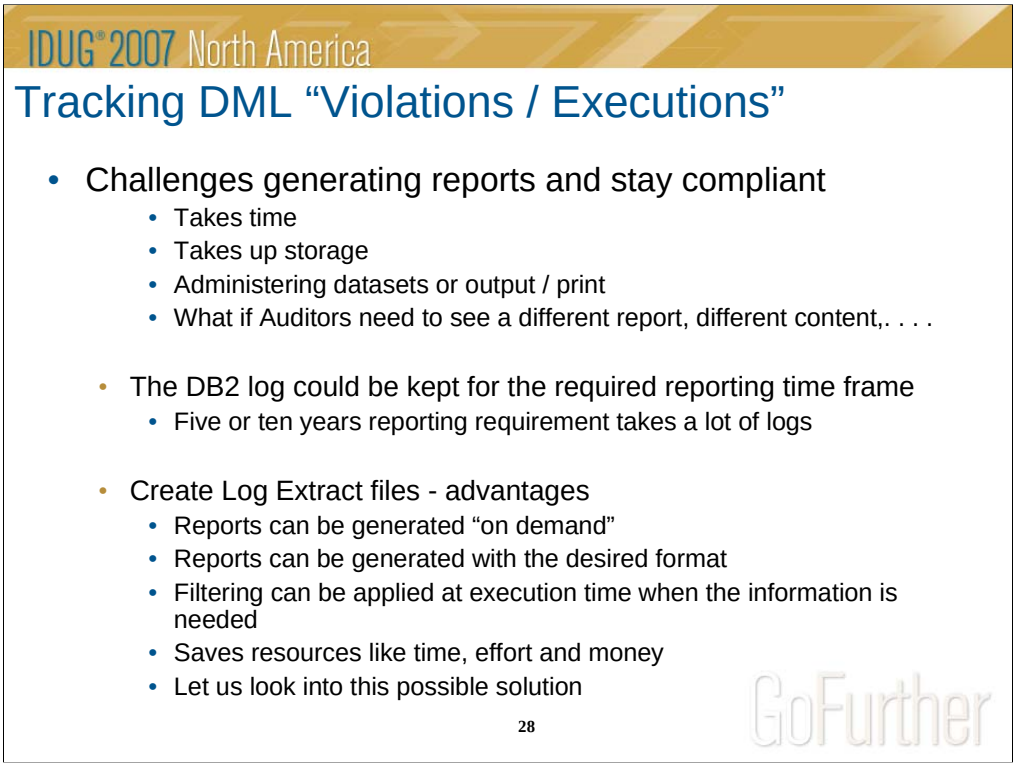
1)Primary auth-id

2)Plan

3)Committed or rolled back

4)Timestamp

Generating REDO-SQL reports/files will not satisfy the need to monitor the primary authorization-id and the plan as well as the timestamp.



Creating reports on a daily basis or every time an archive log is created does however generate some challenges. It takes time, CPU, storage and not to mention administration of all the generated reports. Also – imagine the requirement from the auditors to provide a different report !! In order to satisfy this kind of demand – the DB2 logs could be saved for a longer period, but some regulations require 5 or 10 years reporting, which basically makes it impossible to keep DB2 logs.

To accommodate the need for flexible reporting and also consider the need to retain DB2 logs, one issue to consider is to create "mini logs". Unicenter Log Analyzer provides the ability to create these log extracts including filter specifications like it was illustrated in the previous examples. The idea is to extract the log-records matching the filter specifications and store this information in a log extract file. When there is a need to generate a report, Unicenter Log Analyzer can read the log extract files instead of the active/archive log – and generate the report format required, and by then save time, resources and efforts generating the reports every day.

Let us look into how this works in reality.

May 6–10, 2007

San Jose Convention Center

San Jose, California, USA

Challenges Retaining the DB2 Log

IDUG® 2007 North America

GoFurther

Retaining the DB2 Log

- The DB2 log holds almost all the information we need to stay compliant
- For most sites - keeping the DB2 Log for the required reporting time might not be an option
- Also – the majority of the information contained in the DB2 Log is not needed in order to do audit
  - Checkpoint records
  - Pageset allocation and summary records
  - Exception statuses
  - Backout information
  - Index updates
  - So – maybe 70% of the log isn't needed to stay compliant

As already mentioned, the DB2 log holds all the information related to DML statements we need to conform to the regulatory requirements. Since it might not be possible to retain all the logs needed for the reporting period, extracting the needed information into "mini logs" seems to be a viable solution. Some people might wonder how much space is needed to create these log extracts – and if it's worth the efforts, so let's cover why the extract files might be much smaller compared to the real DB2 logs, since a lot of the information in the DB2 log is not needed to stay compliant:

1) We only need the log-records satisfying the filter specifications, which probably only will be a smaller subset of the tables etc.

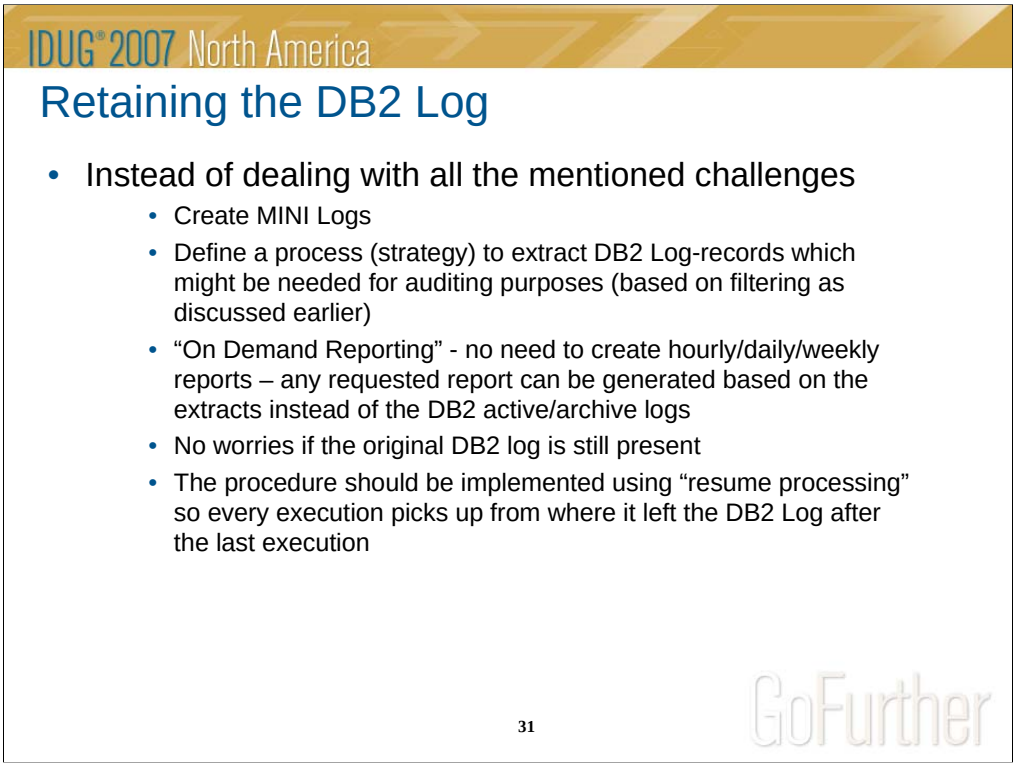
2) We don't need all the information DB2 stores as part of checkpoints, summary records, pageset allocation, exception statuses etc.

3) Every update, delete and insert will have a REDO record for forward recovery purposes – this is what is needed for the extract

4) Every update, delete and insert will also have an UNDO log record in case the transaction will need to be rolled back or is abending – we do not need this information in the extract file.

5) Even though Index Image Copy is not enabled or used, DB2 still logs index updates, which might be a significant amount of logging.

Bottom line – we might need a few percentages of the log for reporting services, making the extract file a great solution.



This can be considered the "cook book" to deal with the challenges of retaining the DB2 log :

1) The solution is to create the "mini logs" / log extracts. Consider using date/time as the log-extract file so it's easier to locate when needed
2) Define the filter specifications for which objects, users, plans etc. need to be included/excluded
3) Schedule a job to extract the needed log-records on a periodic basis starting where the past execution left off.

Implementing this process, it is now possible to generate the needed reports when required, and there is no need to worry about the original DB2 logs – whether they still exist or they have been re-used.

The next slides will illustrate how Unicenter Log Analyzer can be used to automate this process without any manual intervention.

Retaining the DB2 Log

- Part of the automation is to create unique names for the “mini logs” and the optional reports
  - Use date/time so it’s easy to identify which “mini logs” to use later once the need for reporting arises

```
LAOS1 R11.5  ------- Process Log - Output Specifications ------- 01-22-07 17:11

LOG EXTRACT FILE SPECIFICATIONS
  Extract Scope  ==> A   ( F - Filter log data, A - Extract all data )
  Extract DSNAME ==> 'RASST02.SA1G.LOGEXT.D%DATE..T%TIME'
  Disposition    ==> NEW ( SHR , OLD , NEW , RPL )
  New Allocation ==> N   ( Y - Specify allocation info, N - Use current values)

  Control DSNAME ==> 'RASST02.SA1G.CNTL.D%DATE..T%TIME'
  Disposition    ==> NEW ( SHR , OLD , NEW , RPL )
  New Allocation ==> N   ( Y - Specify allocation info, N - Use current values)

REPORT FILE SPECIFICATIONS
  Destination    ==> D   ( D - Dataset, S - Sysout, T - Terminal )
  Dataset Name   ==> 'RASST02.SA1G.REPORT.D%DATE..T%TIME'
  Disposition    ==> NEW ( SHR , OLD , NEW , RPL )
  New Allocation ==> N   ( Y - Specify allocation info, N - Use current values)
            Or...
   Sysout Class   ==> A   Form ==>            Destination ==>
```


The EXTRACT DSNAME is the “mini log”. This scenario is using the subsystem-id and date/time as part of the name so it’s easier to locate the log extract files when needed.

The CONTROL DSNAME is associated with the “mini log” and holds information about the “mini log”.

The Dataset Name for the REPORT file will be the actual report generated based on the filter specifications applied (we covered this some slides back).

Retaining the DB2 Log

- Once the scope is defined (what to extract / which log-records to keep), the process is automated and reports can be generated whenever needed – even though the DB2 Log “is gone”

```
SSID     = (SA1G)
STRATEGY = (SA1G,RASST02,AUDITUS2,SAVE)
RESUME   = (2)
LOGSRC   = (GROUP)
OBJSRC   = (CATALOG)
COPYSRC  = (CATALOG)
DYNSORT  = (DSNUM(6),SPACE(100,100),MAINSIZE(4000),MSG(
GENUNIT  = (SYSDA)
RPTLINES = (60)
S99WAIT  = (YES)
RESOLVUR = (END)
WORKLOAD = (SMALL)
DMLREPT  = (LEVEL    (DETAIL)  ,ROLLBACK (INCLUDE)
           ,CATALOG  (EXCLUDE) ,DSNDB01  (EXCLUDE)
           ,ORDERBY  (URID)
           ,INCLUDE  (AND
                     ,TABLE (%.EMP
                             ,PS12."%" ,SAPR3."%"
                             ,PTI."%" ,PROD."%" )
                      ,AUTHID("SYS%"    , "DBA1%"
                      ,RASST02,RASST01,RASST03   ) )
            ,DISCARDS (0) )
EXTRACT  = (ALL)
```

Control cards generated based on the Specifications described when the Process/strategy was defined.

Resume: how many hours to process (make sure the job runs within the time Frame in order to keep up)

Data Sharing: single members or the entire group

Which tables, databases, user-id’s etc. to include/exclude from the Log Extract process

GoFurther

This is the control cards generated by Unicenter log Analyzer to automate the log extract process, based on the filter specifications etc.

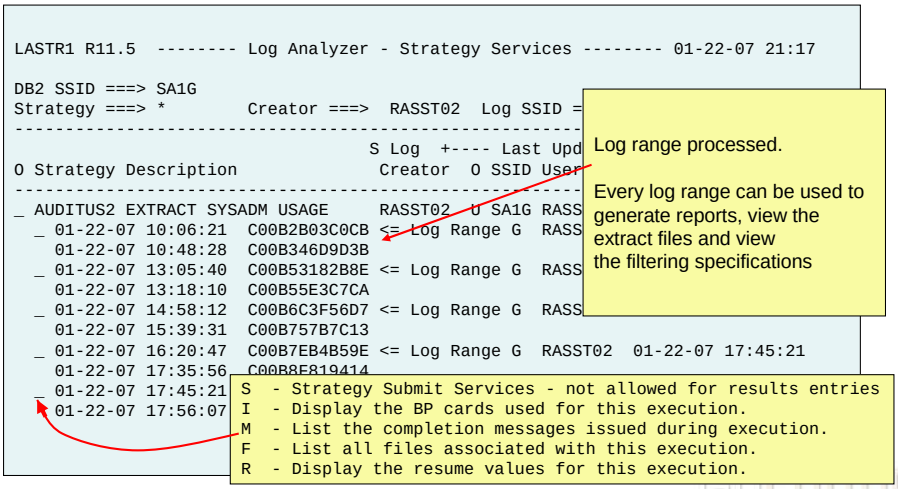
RESUME=2 is how you can control how many hours ahead the job should read the DB2 logs from the past execution. Some users might want to generate the log extract file once a day, so RESUME=24 should then be used.

Remember that each member in a Data Sharing environment has it’s own logs, so a decision needs to be made whether each member should have its own extract, or one covering the entire group is sufficient. It is probably easier to generate one log extract for the entire group so it’s not necessary to modify/add this procedure when members are added/removed.

Finally all the filter specifications can be viewed – which tables are included etc.

Retaining the DB2 Log

- Every strategy / Log Extract can be viewed including log-ranges extracted - and reports can be generated



34

Every time the “mini log” / extract file generation has been executed, the result can be viewed online.

Each execution has the following information: The start date and time and the first LOG-RBA satisfying the filter specifications and the end date and time as well as the end LOG-RBA.

All this information is stored in a DB2 table, so as long as image copies are being executed, there is no danger the information will get lost.

Once there is a demand to generate a report for auditing purposes (or what ever the reason is), simply identify the needed period and use the EXTRACT FILES associated to generate the reports needed (or even REDO information if so desired).

Since the execution details are stored in DB2 tables, it is possible to monitor if anyone have manipulated the results (since updates to these tables also are recorded in the DB2 log).

Retaining the DB2 Log

- Considering a “non static environment”
  - Tables get created
  - Tables get dropped
  - Tables get dropped and re-created
  - User-id’s come and go
- Combining ACM “on demand table definition” covered earlier with the “mini logs” maintained in this section
  - Every table which existed at one point-in-time can be reported
  - Every user-id existing at any point-in-time can be reported
  - Re-created tables and dropped tables can be tracked

35

Most environments are not very static – constantly objects are created, altered, dropped – and re-created. This can provide some challenges when there is a need to report on what happened to a specific table e.g. two years ago, and that table does not exist anymore.

Combining the “Table Definitions On Demand using ACM” with this “mini log” approach will really solve this issue. You will always be in a position to find a specific table definition back in time and then report on DML statements using the log extract.

Even tables which have been dropped and re-created can provide a challenge, since DB2 does not log the table name – but the internal ID’s. Again – using this combination of the two scenarios will provide the ability to also solve this issue.

May 6-10, 2007

San Jose Convention Center

San Jose, California, USA

Not EVERYTHING is in the DB2 Log

IDUG® 2007 North America

GoFurther

Not Everything is in the DB2 Log

- SMF Audit
  - SMF traces are needed in order to report certain violations or accesses
  - Unicenter Log Analyzer provides the ability to specify what should be reported and automatically starts up the necessary traces
  - Reports can then be generated when the SMF records are written

IFCID	Report Title
141	Explicit GRANTs and REVOKEs
142	CREATE, ALTER, and DROP operations
140	Access Attempts Denied
144	First access to an object
143	First Attempted change of an object
55,83,87	Assignment or change of an SQLID
23,24,25	Utilities executed
145	BIND time SQL information
4	Start Trace Information
5	Stop Trace Information

Unfortunately not everything which might be needed to report on can be found in the DB2 log. However – anything which might be needed to report on can be found in SMF-records IFCID's.

Some of the issues which cannot be satisfied from reading the DB2 log are:

1) When someone gets a denied attempt (SQL-551 etc.)

2) When someone assigns a secondary auth-id

3) Some start and stop commands

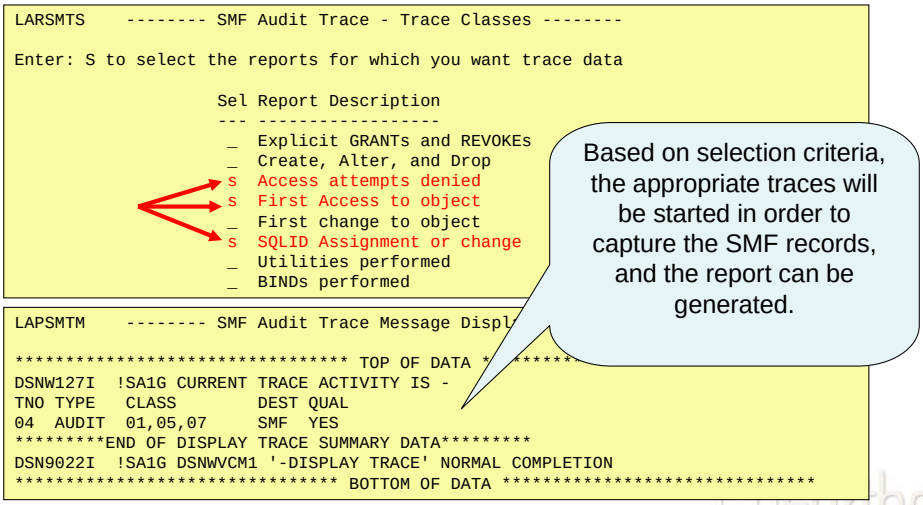
Also – some very sensitive tables might be altered with the AUDIT attribute so it can be tracked exactly who tried to access these tables.

Traces can be started to capture the needed information and then reported on.

Unicenter Log Analyzer can assist in some of these reporting needs – please see the following slides.

Not Everything is in the DB2 Log

- Specify what needs to be tracked / reported



Using Unicenter Log Analyzer SMF Audit feature, simply select the reports needed. Once the reports have been selected, the necessary trace classes will be started based on the filter specifications entered and remain active until stopped.

Not Everything is in the DB2 Log

- If needed - the reports can be generated using filtering so not EVERY user / object etc. is being reported

```
            SET CURRENT SQLID ACTIVITY IFCID = 55 (SET SQLID)

DATE       TIME     SSID PRIMARY  CORREL.  CONNECT  PLAN     PREVIOUS NEW      S
                         AUTH-ID  ID       NAME     NAME     SQLID    SQLID    T
2007-01-11 14:20:03 SA1G RASST02  RASST02  DB2CALL  LAPO1150 RASST02  STEEN    X
2007-01-11 14:51:51 SA1G RASST02  RASST02  DB2CALL  RQPA1150 RASST02  STEEN    X
2007-01-11 14:52:05 SA1G RASST02  RASST02  DB2CALL  RQPA1150 RASST02  STEEN    X
2007-01-11 14:54:25 SA1G RASST02  RASST02  DB2CALL  RCUU1150 RASST02  STEEN    X
2007-01-11 14:58:57 SA1G RASST02  RASST02  DB2CALL  RQPA1150 STEEN    IDUG002  X

--------------------------------------------------------------------------------
                                    AUTHORIZATION FAILURES
 NO RECORDS FOUND
                                              SMF AUDIT REPORTER
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
                                    FIRST ATTEMPTED ACCESS
 NO RECORDS FOUND
                                              SMF AUDIT REPORTER
--------------------------------------------------------------------------------
```


The captured SMF records will be stored in the SMF-datasets which then can be used to generate the reports.

This scenario has requested to generate reports for:

1) Authorization-id / SQL-ID reassignments
2) Report on authorization-id failures to see who tries to access objects where inappropriate authorization exist
3) Report on the first access to certain objects.

In this case only SQL-ID re-assignments were found.

May 6-10, 2007

San Jose Convention Center

San Jose, California, USA

Anything else to Track and Report

IDUG® 2007 North America

GoFurther

Anything else

- Anything else beyond what has been covered can be tracked as well
  - SMF-records / IFCID records
    - The DB2 Administration Guide SC18-7413-00 is a great resource to get details about IFCID's and how to use them
    - IBM macros another useful place to look DB2.SDSNMACS(DSNDQWxx where xx : 00-04)
  - IDB2 IQL requests
    - Use Unicenter CA-Insight for DB2 for z/OS to create any report needed
    - Starting the requests, Insight will start the appropriate trace classes in order to get the information needed

41

As mentioned in the previous section, not everything can be found in the DB2 log, but it might be necessary to use SMF traces to capture these events.

IBM provides a wealth of information to assist in finding the appropriate IFCID's to trace in order to report on the needed events.

If Unicenter CA-Insight for DB2 for z/OS is installed, it is possible to create your own reports using IQL (Insight Query Language). Simply specify the information needed (keywords), and Insight will make sure the appropriate traces are started and reports generated either online or batch.

Session: J10

DB2 Auditing and the Need for NO Secrecy

Steen Rasmussen

CA

steen.rasmussen@ca.com

GoFurther