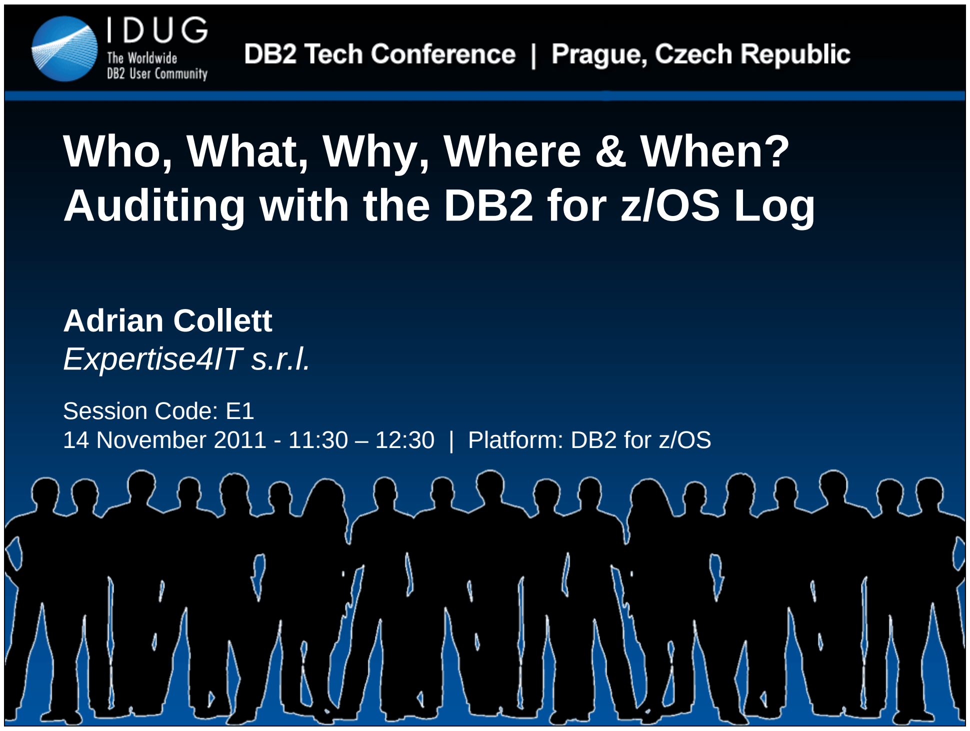


DB2 Tech Conference | Prague, Czech Republic

Who, What, Why, Where & When? Auditing with the DB2 for z/OS Log

Adrian Collett
Expertise4IT s.r.l.

Session Code: E1
14 November 2011 - 11:30 – 12:30 | Platform: DB2 for z/OS

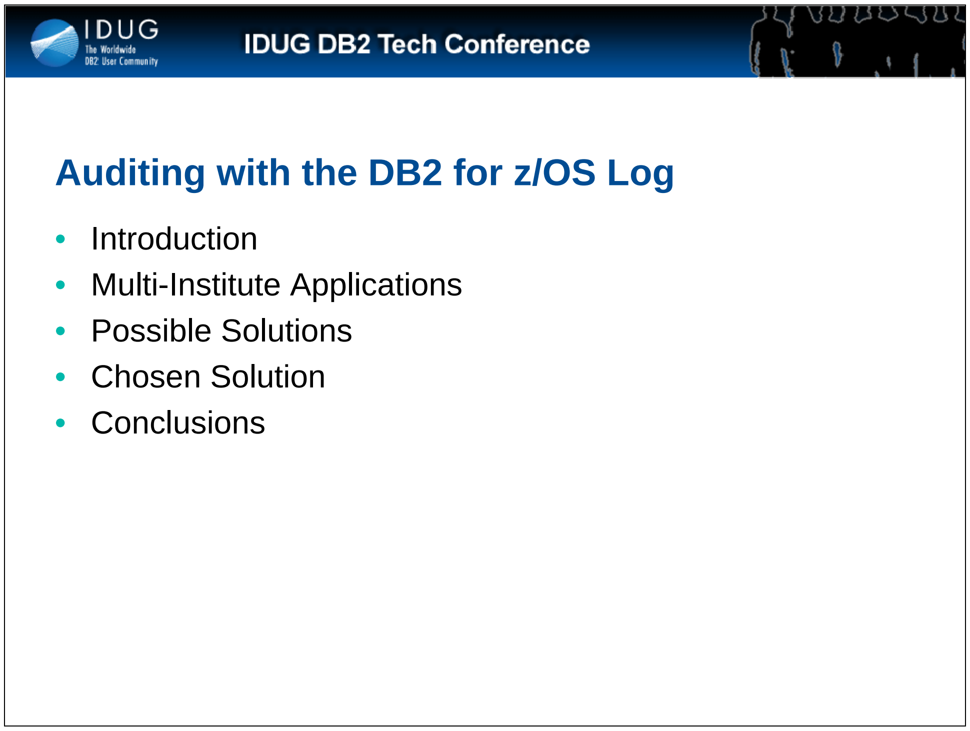
Do you have “Separation of Duties”? Does your Auditor have all the info that they require? Do they really know what they want? Or do they expect you, the humble DBA, to wave a magic wand and immediately produce detailed reports of the Who, What, Why, Where & When of everything?

Is Database Auditing gradually becoming a big problem for DB2 for z/OS DBAs?

This presentation will examine just one topic of database auditing: data modification. It will discuss the various options available to the DB2 for z/OS DBA, their relevance to the main standards and their impact on normal DBA operations such as performance and recovery.

Through a detailed Customer Case Study it will highlight how the use of the DB2 Log can help satisfy, in part, these requirements.

By the end of the session you will have an understanding of the basic data modification auditing requirements, the various options available and, more importantly, you will also be able to decide which ones are appropriate for your needs.



IDUG DB2 Tech Conference

Auditing with the DB2 for z/OS Log

- Introduction
- Multi-Institute Applications
- Possible Solutions
- Chosen Solution
- Conclusions

Objectives:

•Understand the basic requirements for Auditing data access & data modification on DB2 for z/OS;

•Understand the various options that are currently available to the DB2 for z/OS DBA to satisfy these requirements, their completeness with regards to the various standards and their impact on normal DBA operations;

•Understand how the use of the DB2 Log can be used to satisfy in part these requirements and, more importantly, understand which requirements it does not satisfy;

•Understand the implications and complexities of using the DB2 Log, Data Capture, Data Compression, Multi-Institute Applications and the problems inherent in satisfying the "Separation of Duties" auditing requirement;

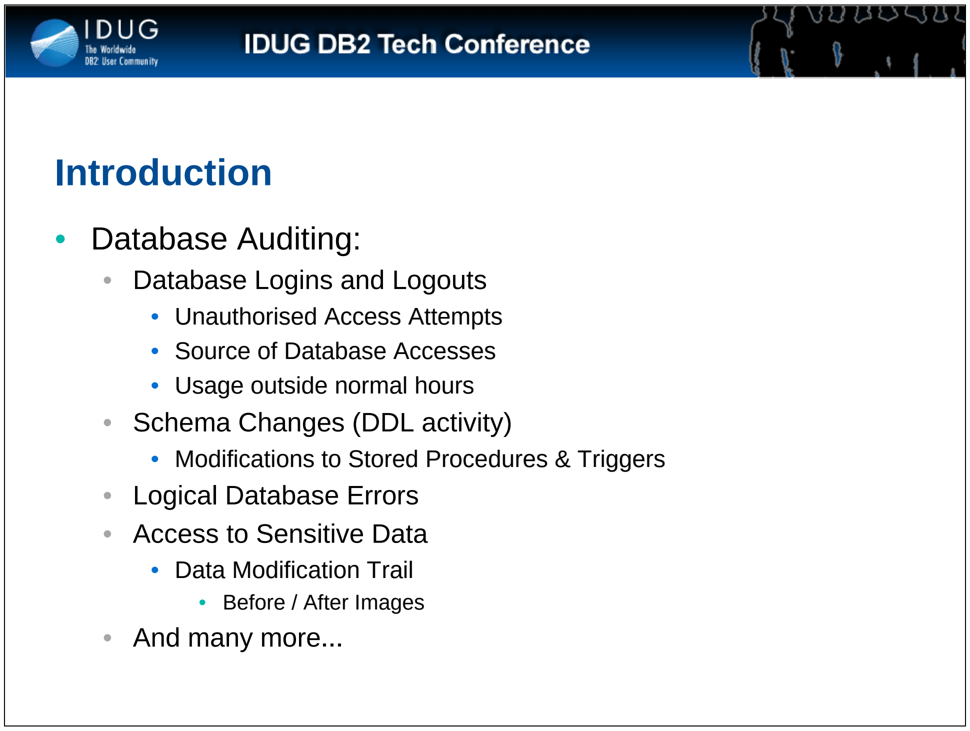
•Be able to understand and identify which Database Auditing technique best fits their company's requirements.

Disclaimer

I am ***NOT*** an Auditor ☺ The goal of this presentation is to share my experience of using the DB2 for z/OS Log as a source for Auditing purposes. It is not intended as a guide to Auditing. This presentation is ***NOT*** intended to provide regulatory, legal or other advice regarding Auditing, Data Governance and/or Data Security. These areas should be addressed by qualified regulatory and/or data security/governance professionals.

Auditing with the DB2 for z/OS Log

- Introduction
- Multi-Institute Applications
- Possible Solutions
- Chosen Solution
- Conclusions


IDUG DB2 Tech Conference

Introduction

- Database Auditing:
  - Database Logins and Logouts
    - Unauthorised Access Attempts
    - Source of Database Accesses
    - Usage outside normal hours
  - Schema Changes (DDL activity)
    - Modifications to Stored Procedures & Triggers
  - Logical Database Errors
  - Access to Sensitive Data
    - Data Modification Trail
      - Before / After Images
  - And many more...

Database Auditing is a vast and complex subject covering all aspects of Database usage, the slide highlights just some of the areas involved.

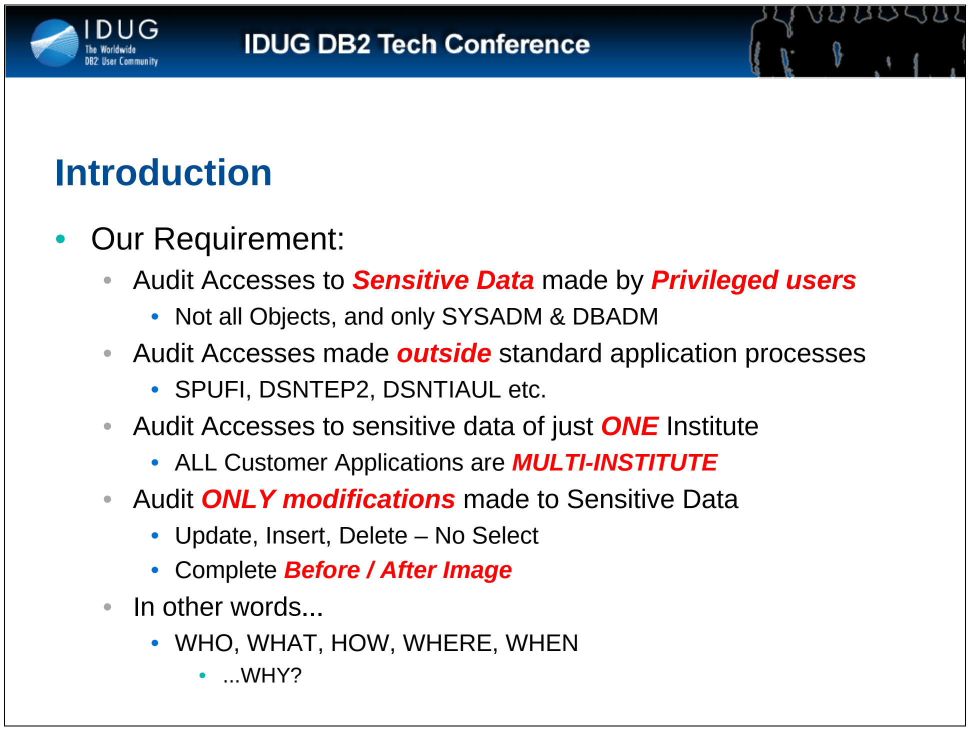
We need to know who is accessing the database, when and from where they are accessing it as well as any unauthorised access attempts.

We also need to track who is making modifications to the structure of the Database. This is especially important when we consider that we may have important business logic “hidden” inside database objects such as Stored procedures and Triggers.

Likewise we also need to Audit any Logical errors within the applications.

Another important area of Database Auditing is that of auditing accesses to sensitive data within the Database. In many cases this may be limited to just identifying who is accessing the data, in others it may require a detailed Data Modification Trail involving Before and After Images of every row that is inserted/updated or deleted.

As I said at the start, I am ***not*** an Auditor, so this list is far from complete.


IDUG The Worldwide DB2 User Community — IDUG DB2 Tech Conference

Introduction

- Our Requirement:
  - Audit Accesses to ***Sensitive Data*** made by ***Privileged users***
    - Not all Objects, and only SYSADM & DBADM
  - Audit Accesses made ***outside*** standard application processes
    - SPUFI, DSNTEP2, DSNTIAUL etc.
  - Audit Accesses to sensitive data of just ***ONE*** Institute
    - ALL Customer Applications are ***MULTI-INSTITUTE***
  - Audit ***ONLY modifications*** made to Sensitive Data
    - Update, Insert, Delete – No Select
    - Complete ***Before / After Image***
  - In other words...
    - WHO, WHAT, HOW, WHERE, WHEN
      - ...WHY?

This presentation will concentrate on just one of those aspects of Database Auditing, that of a Data Modification Trail for accesses to sensitive data. We were asked to implement a process to audit accesses to sensitive data made by privileged users. In this context a privileged user was defined as those users with SYSADM or DBADM authority on the sensitive data. So, in practice we had to audit a very small set of users.

The Audit process only had to consider accesses to the data made outside of standard application processes. The customer was confident that all other accesses were satisfactorily audited via other mechanisms such as Job or Application logs.

The customers applications process data for several financial institutions, however, it was only necessary to Audit data of just one Institute.

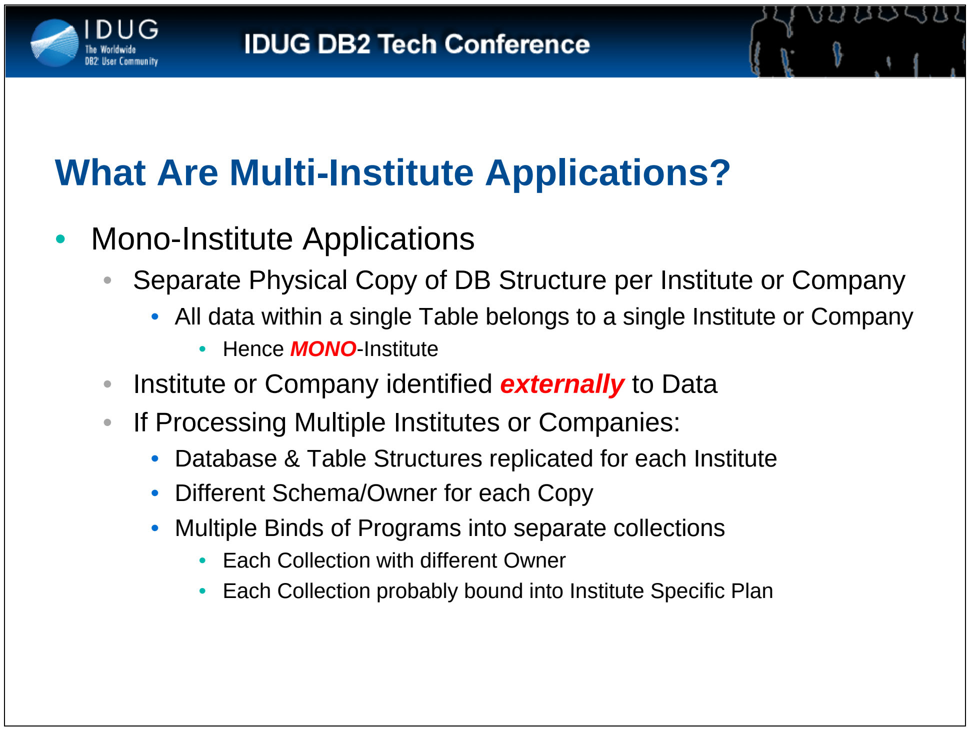
The critical part of the customer requirement, however, was that the process had to provide detailed audit information of every modification made to the data.

In other words the full WHO, WHAT, HOW, WHERE, WHEN of the Update, Insert or Delete.... But not the WHY !!!

This presentation will briefly examine the various options available to us and will then outline in detail our chosen solution.

Auditing with the DB2 for z/OS Log

- Introduction
- Multi-Institute Applications
- Possible Solutions
- Chosen Solution
- Conclusions


IDUG The Worldwide DB2 User Community

IDUG DB2 Tech Conference

What Are Multi-Institute Applications?

- Mono-Institute Applications
  - Separate Physical Copy of DB Structure per Institute or Company
    - All data within a single Table belongs to a single Institute or Company
      - Hence ***MONO***-Institute
  - Institute or Company identified ***externally*** to Data
  - If Processing Multiple Institutes or Companies:
    - Database & Table Structures replicated for each Institute
    - Different Schema/Owner for each Copy
    - Multiple Binds of Programs into separate collections
      - Each Collection with different Owner
      - Each Collection probably bound into Institute Specific Plan

Before going any further, it is important to highlight what I mean by Mono or Multi-Institute Applications and their respective impact on auditing.

Although I call them ***Mono***-institute applications they can actually process data for several institutes or companies etc.

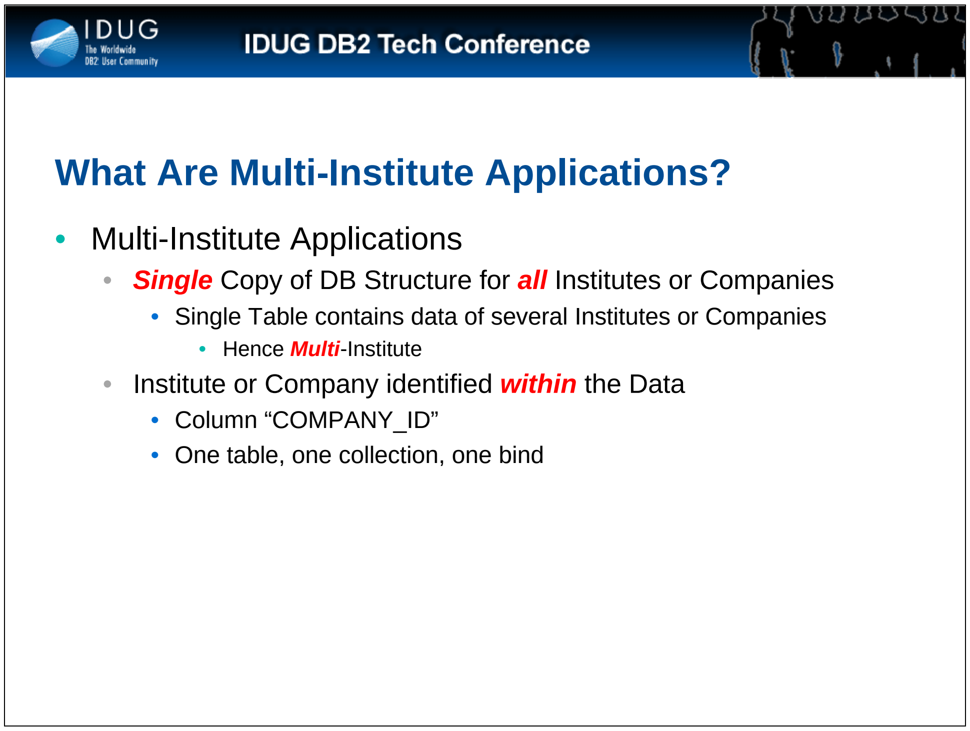
The major characteristic is the fact that they have a separate, physical Database/Table structures for each separate Institute or Company being processed by the application.

Therefore, all the data in each single table is related to one and one only institute or company.

If we need to process more than one institute, then the Database/Table structures are replicated for each Institute..

Normally this is done by identifying each copy of the structure via a different OWNER of the tables.

The application programs are then bound into separate collections for each institute or company. And probably each collection is bound to an Institute-specific Plan.



IDUG DB2 Tech Conference

What Are Multi-Institute Applications?

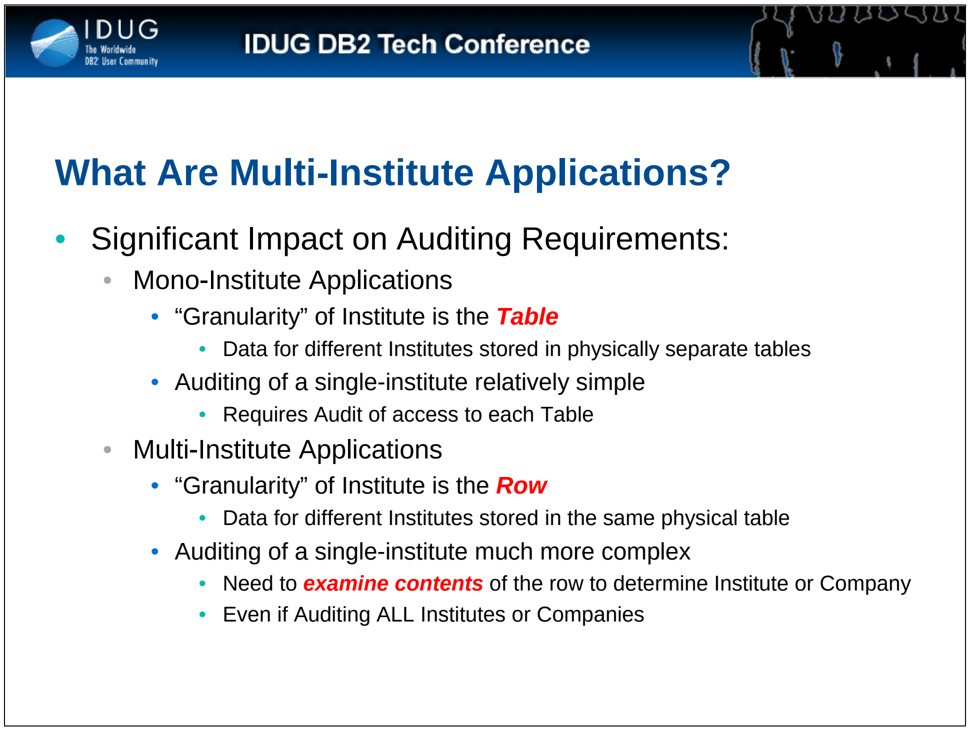
- Multi-Institute Applications
  - ***Single*** Copy of DB Structure for ***all*** Institutes or Companies
    - Single Table contains data of several Institutes or Companies
      - Hence ***Multi***-Institute
  - Institute or Company identified ***within*** the Data
    - Column “COMPANY_ID”
    - One table, one collection, one bind

Now let’s look at what I mean by a Multi-Institute Application:

Again, just as with a Mono-Institute Application, a Multi-Institute Application can process data for either a single institute or for several institutes or companies.

However, with a Multi-Institute Application, the identification of the Institute being processed is done on the basis of the contents of the data. This is normally done via a column which identifies the company or institute: COMPANY_ID for example.

With this type of Application Architecture, we only have one physical table for our data and therefore only one Bind into a single collection is required.


IDUG The Worldwide DB2 User Community

IDUG DB2 Tech Conference

What Are Multi-Institute Applications?

- Significant Impact on Auditing Requirements:
  - Mono-Institute Applications
    - "Granularity" of Institute is the ***Table***
      - Data for different Institutes stored in physically separate tables
    - Auditing of a single-institute relatively simple
      - Requires Audit of access to each Table
  - Multi-Institute Applications
    - "Granularity" of Institute is the ***Row***
      - Data for different Institutes stored in the same physical table
    - Auditing of a single-institute much more complex
      - Need to ***examine contents*** of the row to determine Institute or Company
      - Even if Auditing ALL Institutes or Companies

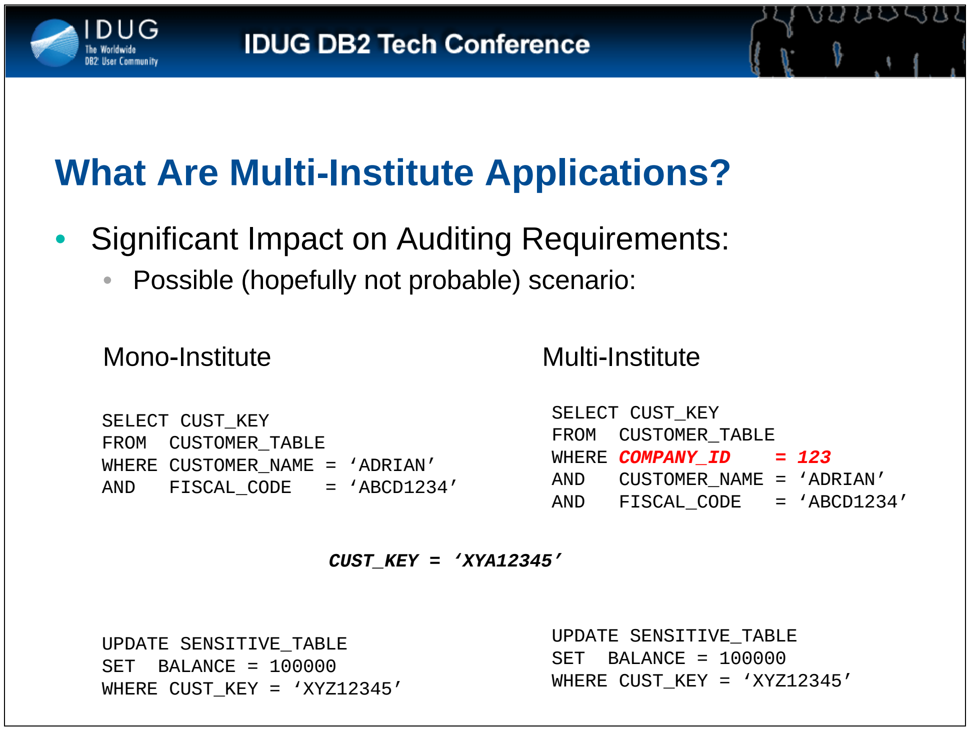
As you can imagine, both of the two different solutions have their pros and cons, and it is not the purpose of this presentation to examine them in detail. However, it is important that we understand the principal characteristics of the two solutions and their ***profound*** implications on Auditing. Especially, as was our case, in the audit of a ***single*** institute.

In fact, in the case of a Mono-Institute Application, the granularity of the institute is at the ***Table*** level: the data for each different institute is physically stored in separate tables from the other institutes. This makes auditing of a ***single*** Institute a relatively simple operation as it is only necessary to audit access to the tables of that particular Institute.

However, in the case of a Multi-Institute Application, the "granularity" of the institute is now at ***ROW*** level: in the same table we have data from several institutes. Therefore, the audit of a ***single*** institute becomes much more complex as now every single row has to be examined in order to establish whether it belongs to the institute being audited.

Also, even if we are auditing ALL institutes within the table, we still need to examine each row, and in particular the identifying column (e.g. COMPANY_ID), in order to identify which institute has been modified.

And as we'll see later, this is not as simple as it seems!



What Are Multi-Institute Applications?

- Significant Impact on Auditing Requirements:
  - Possible (hopefully not probable) scenario:

Mono-Institute

```
SELECT CUST_KEY
FROM   CUSTOMER_TABLE
WHERE CUSTOMER_NAME = 'ADRIAN'
AND   FISCAL_CODE   = 'ABCD1234'
```

Multi-Institute

```
SELECT CUST_KEY
FROM   CUSTOMER_TABLE
WHERE COMPANY_ID    = 123
AND   CUSTOMER_NAME = 'ADRIAN'
AND   FISCAL_CODE   = 'ABCD1234'
```

CUST_KEY = 'XYA12345'

```
UPDATE SENSITIVE_TABLE
SET   BALANCE = 100000
WHERE CUST_KEY = 'XYZ12345'
```

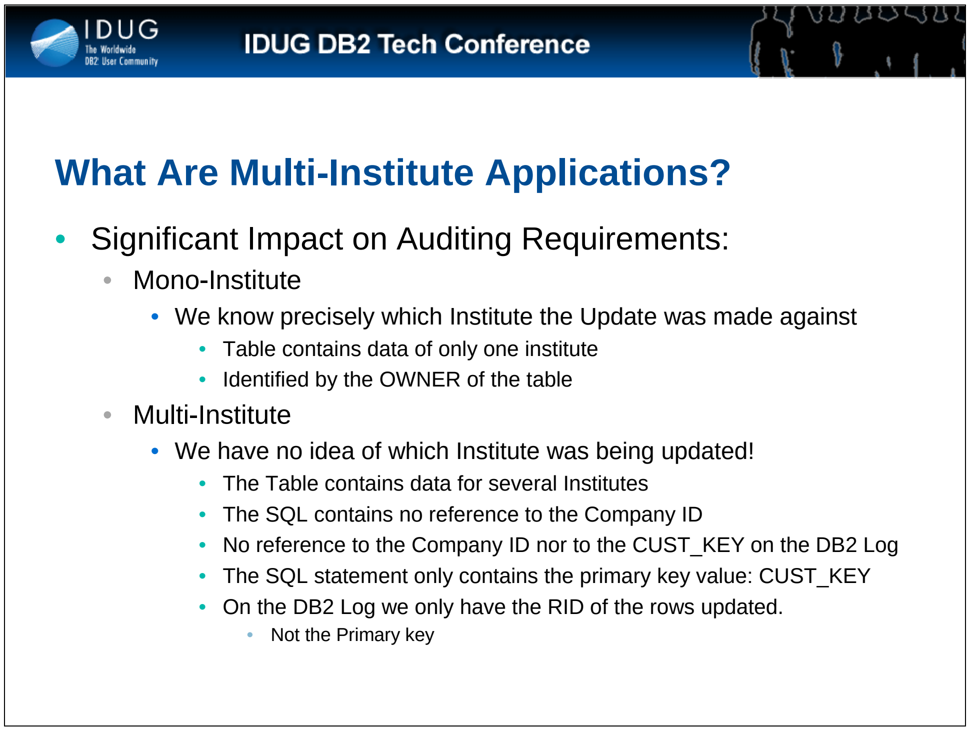
```
UPDATE SENSITIVE_TABLE
SET   BALANCE = 100000
WHERE CUST_KEY = 'XYZ12345'
```

To further demonstrate the auditing implications of Multi-Institute Applications, let's examine the following scenario:

Our application first accesses a customer table to extract a unique key of a particular customer.

Due to the fact that the CUST_KEY is unique to the application and ***not*** to the COMPANY, the subsequent update does not need to specify a predicate for COMPANY_ID.

And hence, the UPDATE has NO reference to which institute has been updated.



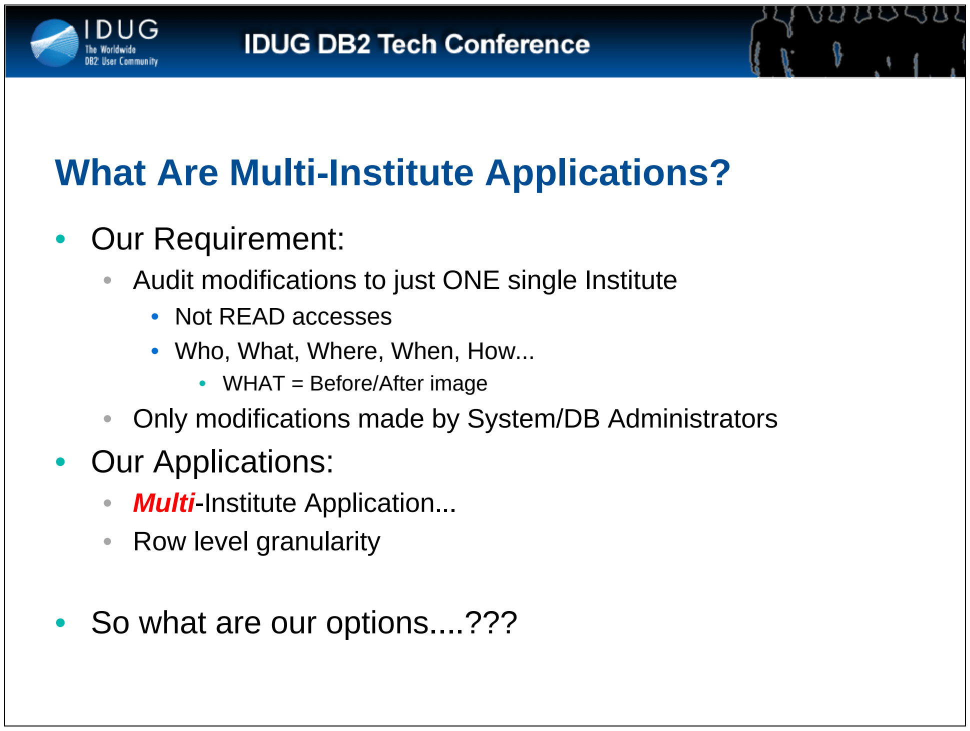
What Are Multi-Institute Applications?

- Significant Impact on Auditing Requirements:
  - Mono-Institute
    - We know precisely which Institute the Update was made against
      - Table contains data of only one institute
      - Identified by the OWNER of the table
  - Multi-Institute
    - We have no idea of which Institute was being updated!
      - The Table contains data for several Institutes
      - The SQL contains no reference to the Company ID
      - No reference to the Company ID nor to the CUST_KEY on the DB2 Log
      - The SQL statement only contains the primary key value: CUST_KEY
      - On the DB2 Log we only have the RID of the rows updated.
        - Not the Primary key

In a MONO-Institute application, we know exactly which Institute or Company the Update is being performed against, as the table contains data of a single Institute.

As you can see, it is extremely complicated to identify the Company or Institute in this scenario due to the fact that the Table contains data of several institutes.

Obviously, one solution is to try and force the use of the COMPANY_ID predicate in all SQL, however, an Audit cannot assume that all the applications are well-disciplined and, of course, even then the predicate is not written on the DB2 Log.


IDUG The Worldwide DB2 User Community

IDUG DB2 Tech Conference

What Are Multi-Institute Applications?

- Our Requirement:
  - Audit modifications to just ONE single Institute
    - Not READ accesses
    - Who, What, Where, When, How...
      - WHAT = Before/After image
  - Only modifications made by System/DB Administrators
- Our Applications:
  - ***Multi***-Institute Application...
  - Row level granularity
- So what are our options....???

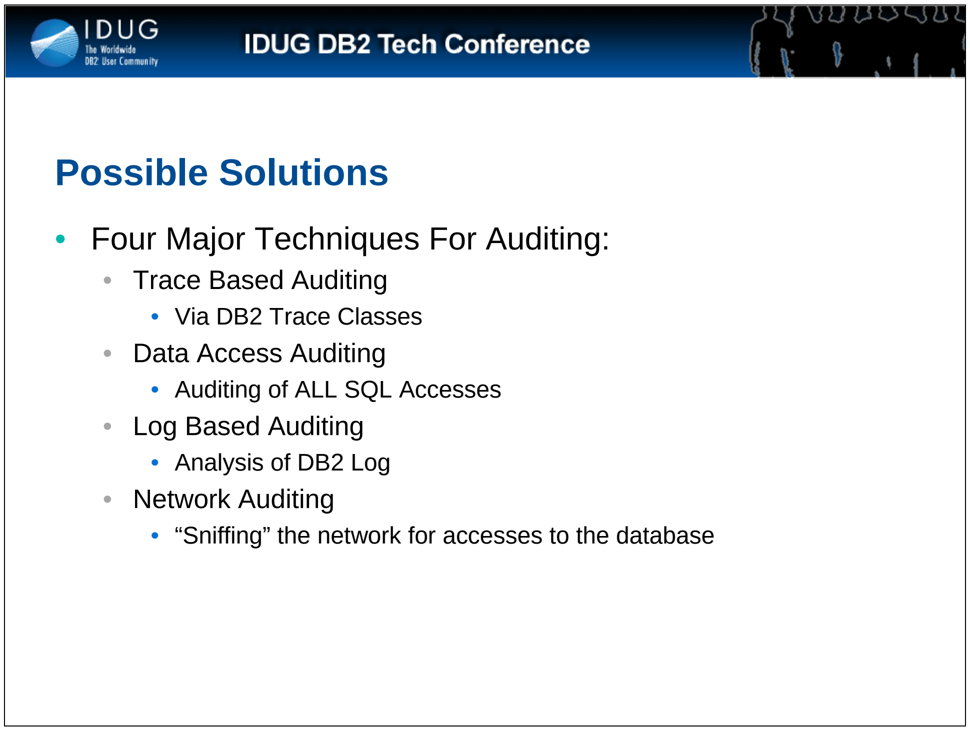
Our requirement was to Audit all modifications made by System and Database Administrators to data for a SINGLE institute, we weren’t interested in Read Accesses.

However all of our applications were all MULTI-Institute Applications!!!

The rest of the presentation highlights the various options that were available to us and how we implemented our chosen solution.

Auditing with the DB2 for z/OS Log

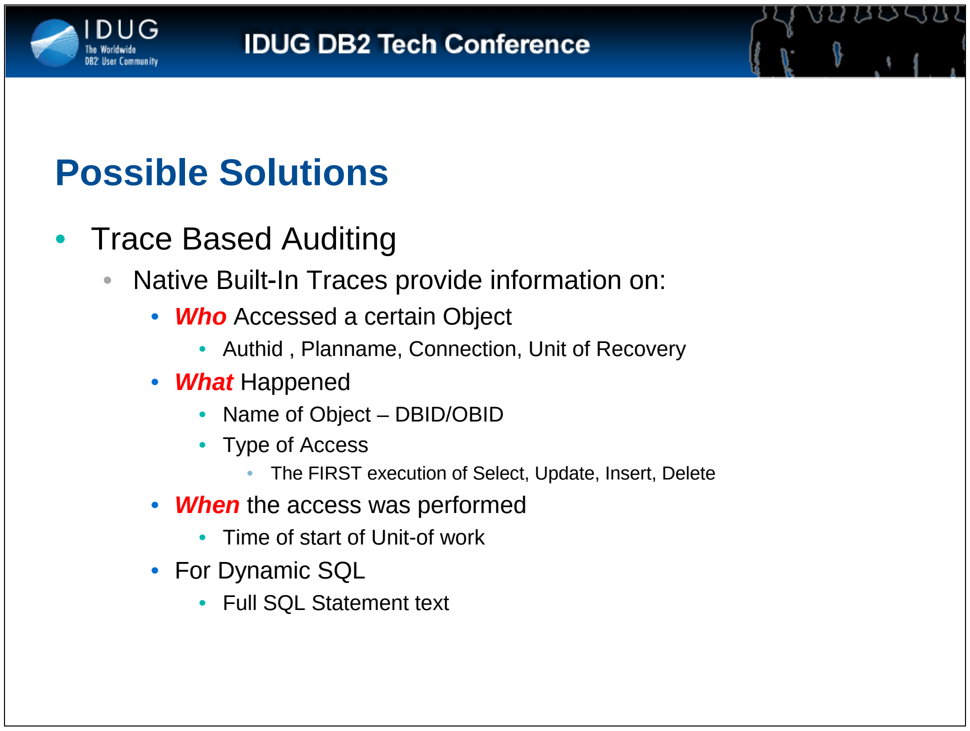
- Introduction
- Multi-Institute Applications
- Possible Solutions
- Chosen Solution
- Conclusions


IDUG DB2 Tech Conference

Possible Solutions

- Four Major Techniques For Auditing:
  - Trace Based Auditing
    - Via DB2 Trace Classes
  - Data Access Auditing
    - Auditing of ALL SQL Accesses
  - Log Based Auditing
    - Analysis of DB2 Log
  - Network Auditing
    - “Sniffing” the network for accesses to the database

There are four major techniques used in Database Auditing, all with their own advantages and disadvantages, so lets take a look at them in detail...


IDUG DB2 Tech Conference

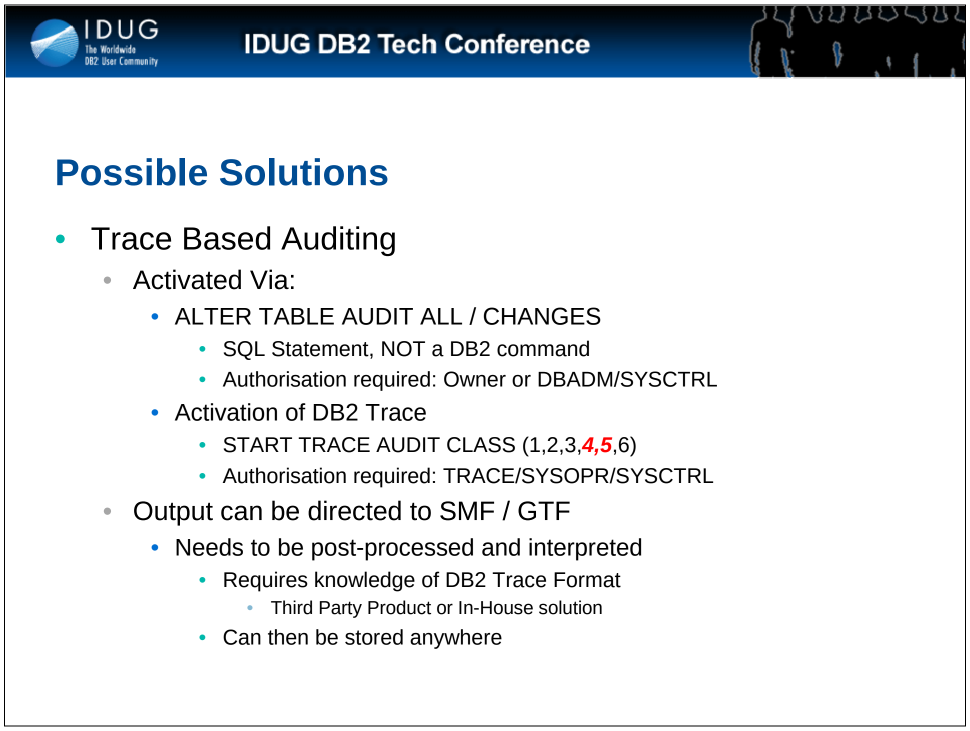
Possible Solutions

- Trace Based Auditing
  - Native Built-In Traces provide information on:
    - ***Who*** Accessed a certain Object
      - Authid , Planname, Connection, Unit of Recovery
    - ***What*** Happened
      - Name of Object – DBID/OBID
      - Type of Access
        - The FIRST execution of Select, Update, Insert, Delete
    - ***When*** the access was performed
      - Time of start of Unit-of work
    - For Dynamic SQL
      - Full SQL Statement text

The first possible solution is Trace Based Auditing, using the native traces of DB2.

The traced data will provide information on WHO accessed the data, WHAT data was accessed and WHEN it was accessed.

For Dynamic SQL it will also provide the full Text of the SQL Statement(V8 only the first 4000 bytes).


IDUG DB2 Tech Conference

Possible Solutions

- Trace Based Auditing
  - Activated Via:
    - ALTER TABLE AUDIT ALL / CHANGES
      - SQL Statement, NOT a DB2 command
      - Authorisation required: Owner or DBADM/SYSCTRL
    - Activation of DB2 Trace
      - START TRACE AUDIT CLASS (1,2,3,***4,5***,6)
      - Authorisation required: TRACE/SYSOPR/SYSCTRL
  - Output can be directed to SMF / GTF
    - Needs to be post-processed and interpreted
      - Requires knowledge of DB2 Trace Format
        - Third Party Product or In-House solution
      - Can then be stored anywhere

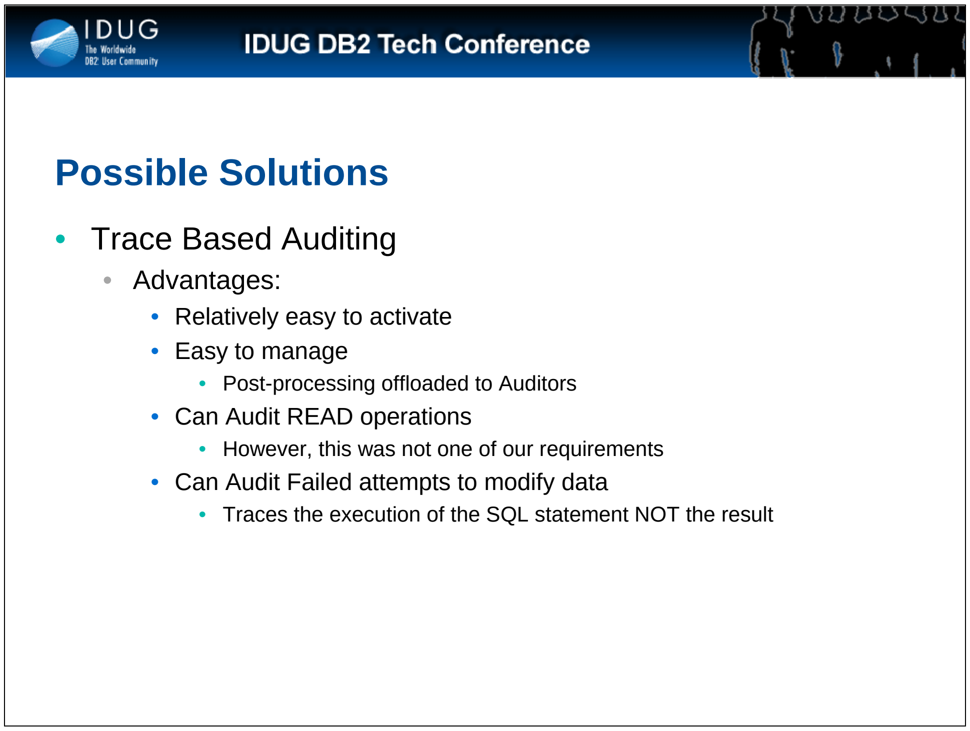
Trace Based Auditing is activated via two separate operations:
The first requires the execution of a simple SQL ALTER TABLE AUDIT ALL / CHANGES statement on all tables that need to be Audited.

Note, this operation is an SQL Statement and not a DB2 COMMAND, and will be traced by Audit Trace CLASS(3) – CREATE,ALTER, DROP. Also, this operation requires ownership of the table , DBADM or SYSCTRL Authority.

The second step in activation of the Audit Trace requires the execution of the START TRACE command; classes 4 & 5 trace SQL accesses to the audited objects. This operation requires either explicit TRACE privilege or SYSOPR/SYSCTRL privileges.

The output generated by the DB2 Trace can be directed to SMF or GTF but then requires post-processing. This post-processing requires detailed knowledge of DB2 Trace records and therefore the only viable solution would really be an ISV Product.

Any In-House solution poses a threat to the Audit process as the person possessing the detailed knowledge of the DB2 Trace records could decide to filter out certain information.

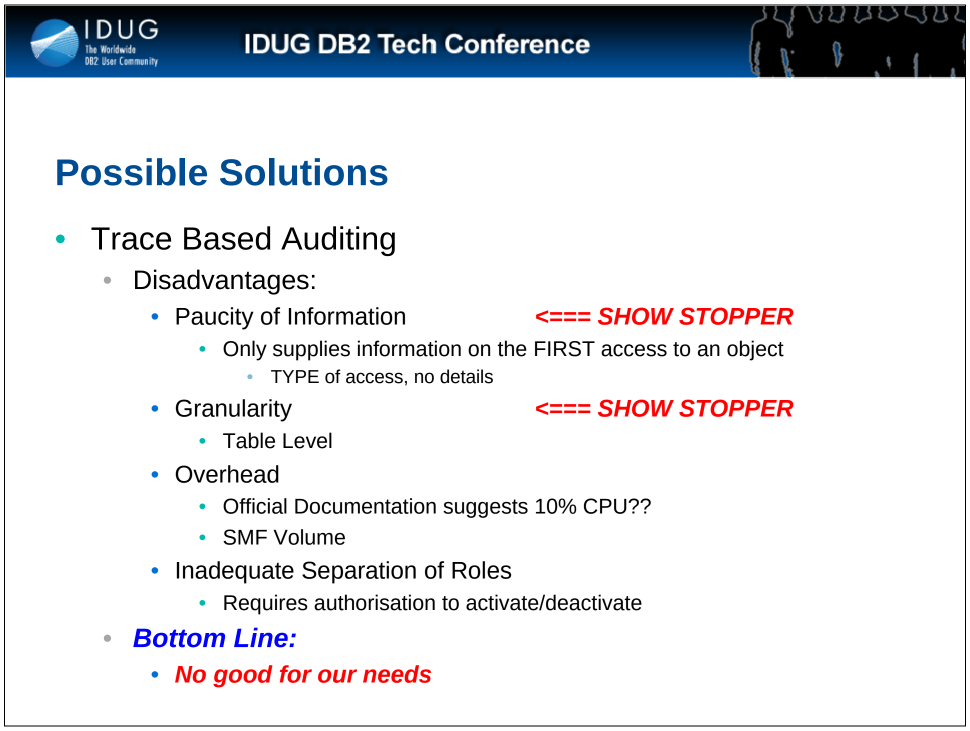

IDUG DB2 Tech Conference

Possible Solutions

- Trace Based Auditing
  - Advantages:
    - Relatively easy to activate
    - Easy to manage
      - Post-processing offloaded to Auditors
    - Can Audit READ operations
      - However, this was not one of our requirements
    - Can Audit Failed attempts to modify data
      - Traces the execution of the SQL statement NOT the result

The Advantages of Trace Based Auditing are that it is fairly easy to activate and manage; for example post-processing can be offloaded to a different platform and control can be delegated to auditors via ISV products.

Also, READ accesses can be audited, although this was not one of our requirements as can FAILED attempts to modify data: the trace audits the execution of the SQL statement independently of whether the SQL statement actually modifies any data or not.


IDUG The Worldwide DB2 User Community

IDUG DB2 Tech Conference

Possible Solutions

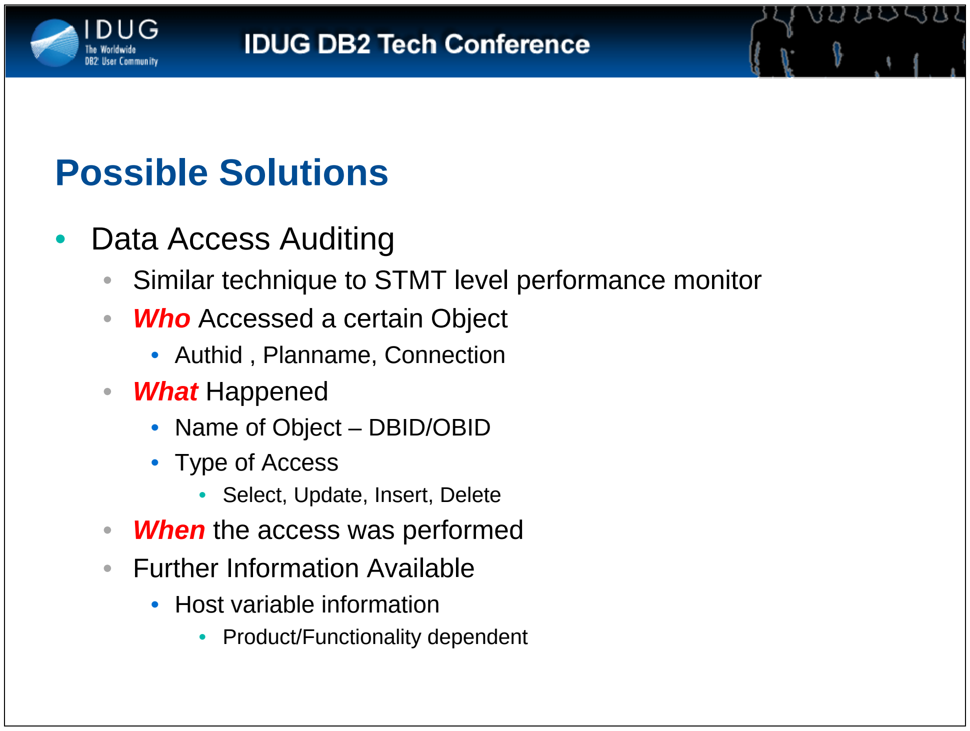
- Trace Based Auditing
  - Disadvantages:
    - Paucity of Information **<=== *SHOW STOPPER***
      - Only supplies information on the FIRST access to an object
        - TYPE of access, no details
    - Granularity **<=== *SHOW STOPPER***
      - Table Level
    - Overhead
      - Official Documentation suggests 10% CPU??
      - SMF Volume
    - Inadequate Separation of Roles
      - Requires authorisation to activate/deactivate
  - ***Bottom Line:***
    - ***No good for our needs***

There are however, several disadvantages to Trace-Based Auditing, the main one being the paucity of information provided by the trace: the trace audits only the fact that a Unit of Work attempted to modify data within an audited object. There is no information on which data, if any, was modified, no BEFORE/AFTER Images, no indication of the number of records modified etc.

Also, and for us this was a show stopper, the granularity of the Audit Trace is at Table Level, and not at ROW level.

Other disadvantages are the Overhead of the Audit Trace, which can, according to the manuals, reach 10% as well as SMF Volume.

And finally, there could be an issue with Separation of Roles as the Trace-Based Audit requires some fairly powerful DB2 authorities which most shops won't want to confer on to a Security Administrator.


IDUG DB2 Tech Conference

Possible Solutions

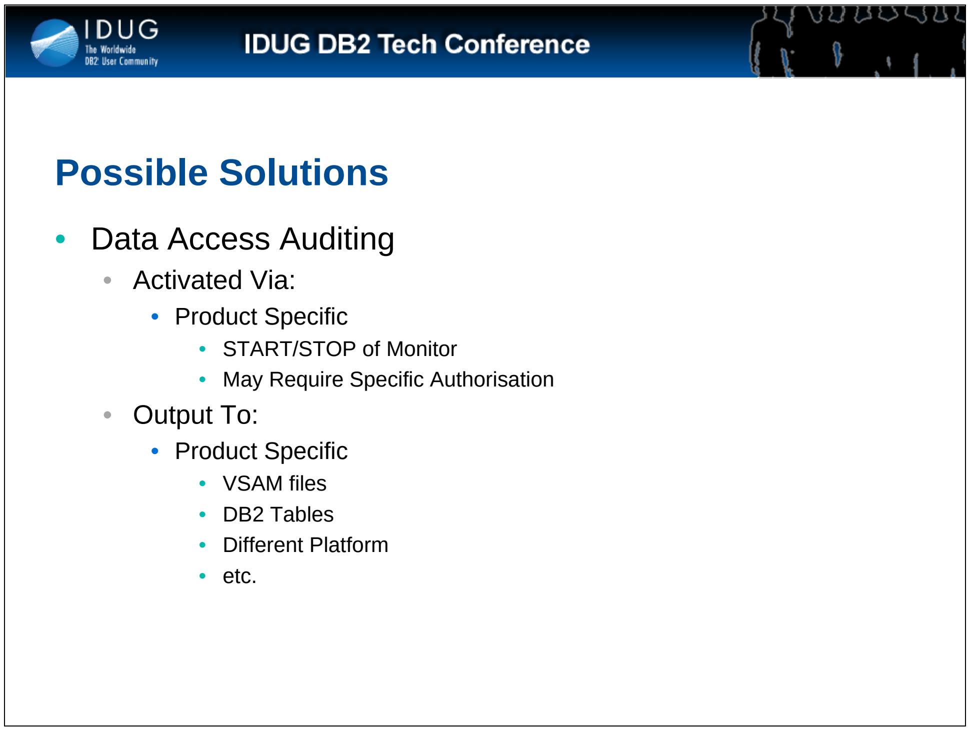
- Data Access Auditing
  - Similar technique to STMT level performance monitor
  - *Who* Accessed a certain Object
    - Authid , Planname, Connection
  - *What* Happened
    - Name of Object – DBID/OBID
    - Type of Access
      - Select, Update, Insert, Delete
  - *When* the access was performed
  - Further Information Available
    - Host variable information
      - Product/Functionality dependent

Another technique is Data Access Auditing, which basically uses the same data which most STMT level performance monitors use.

i.e. All database SQL requests are captured and stored as they are made.

This Auditing Technique requires the use of an ISV Product.

As is to be expected, this Auditing technique can provide the required WHO, WHAT, WHERE, WHEN information and may also provide further information like Host Variable content, but this depends on the ISV product used for gathering the data and on the functionality activated.


IDUG The Worldwide DB2 User Community

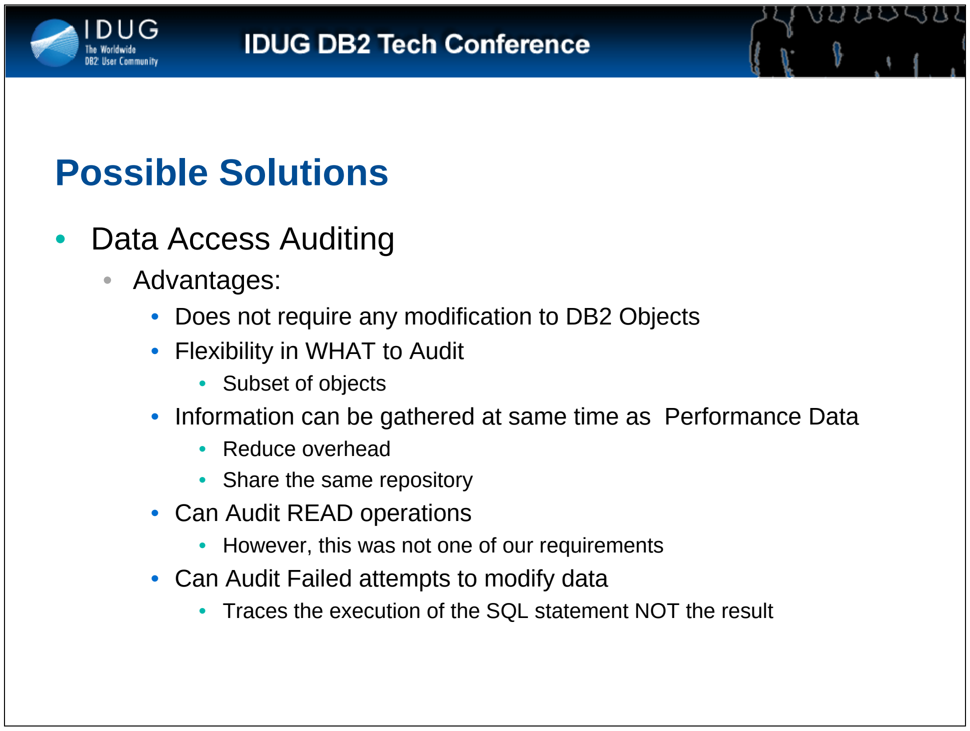
IDUG DB2 Tech Conference

Possible Solutions

- Data Access Auditing
  - Activated Via:
    - Product Specific
      - START/STOP of Monitor
      - May Require Specific Authorisation
  - Output To:
    - Product Specific
      - VSAM files
      - DB2 Tables
      - Different Platform
      - etc.

Activating Data Access Auditing is product specific and is not the subject of this presentation, suffice to say that it will invariably involve the START of a monitor which in turn may require a certain level of authorisation within the DB2 Subsystem.

Again the output generated is purely product specific.


IDUG The Worldwide DB2 User Community

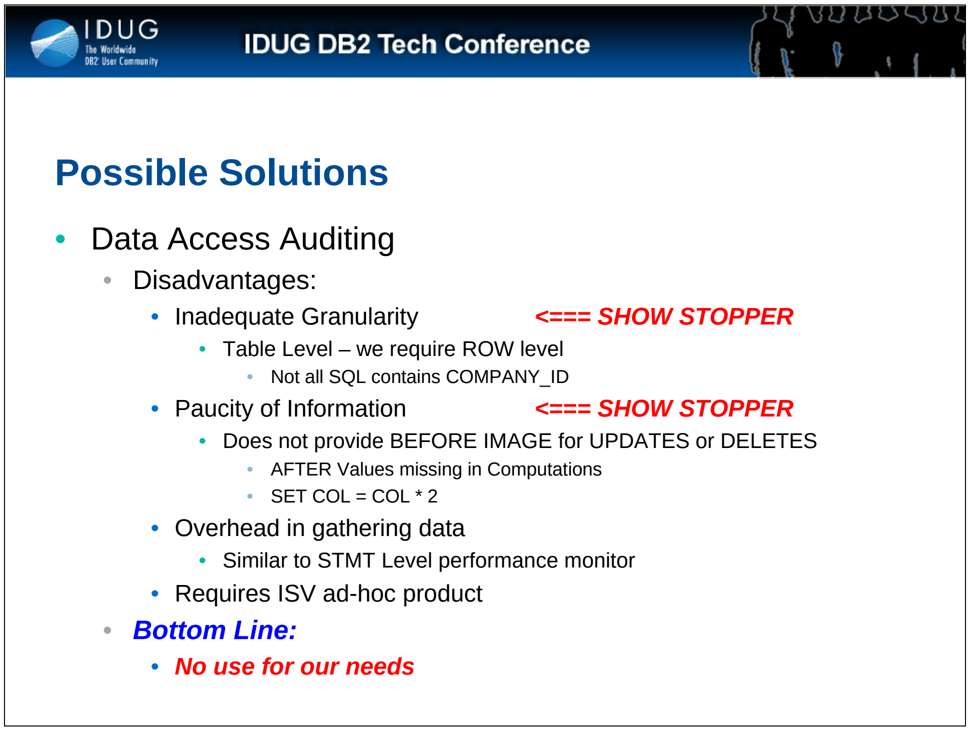
IDUG DB2 Tech Conference

Possible Solutions

- Data Access Auditing
  - Advantages:
    - Does not require any modification to DB2 Objects
    - Flexibility in WHAT to Audit
      - Subset of objects
    - Information can be gathered at same time as Performance Data
      - Reduce overhead
      - Share the same repository
    - Can Audit READ operations
      - However, this was not one of our requirements
    - Can Audit Failed attempts to modify data
      - Traces the execution of the SQL statement NOT the result

The main advantages of Data Access Auditing are that it does not require any modification to the objects being audited and that the audit data can be generated at the same time as the data of a Stmt-Level Performance monitor, thus reducing its overhead.

As with Trace-Based Auditing this can also audit READ accesses and FAILED update attempts.


IDUG DB2 Tech Conference

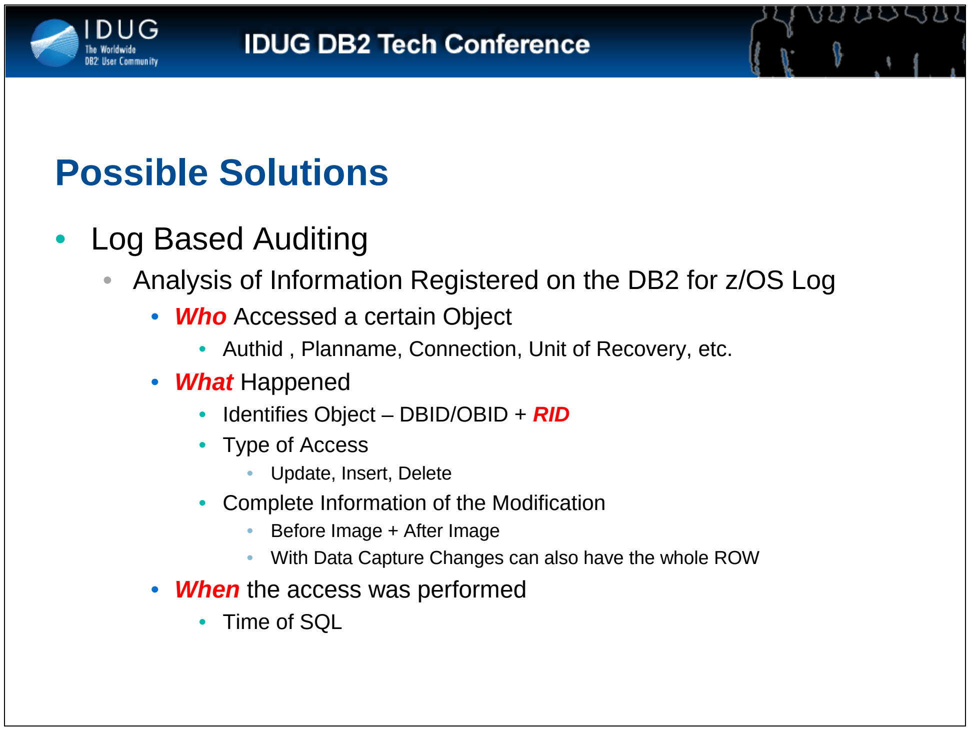
Possible Solutions

- Data Access Auditing
  - Disadvantages:
    - Inadequate Granularity *<=== SHOW STOPPER*
      - Table Level – we require ROW level
        - Not all SQL contains COMPANY_ID
    - Paucity of Information *<=== SHOW STOPPER*
      - Does not provide BEFORE IMAGE for UPDATES or DELETES
        - AFTER Values missing in Computations
        - SET COL = COL * 2
    - Overhead in gathering data
      - Similar to STMT Level performance monitor
    - Requires ISV ad-hoc product
  - ***Bottom Line:***
    - ***No use for our needs***

As with Trace-Based Auditing, the disadvantages of Data Access Auditing are the granularity of the Audited Data, we require ROW level; and the paucity of information provided: we have no BEFORE image, and for operations involving computations we have no AFTER Image, and there is no indication of the number of records modified etc.

Other disadvantages are the Overhead in gathering the data; especially if the ISV product used is NOT the same as the STMT-Level Performance Monitor being used and also if we require host-variable information.

And of course, there may also be a BUDGET related disadvantage!

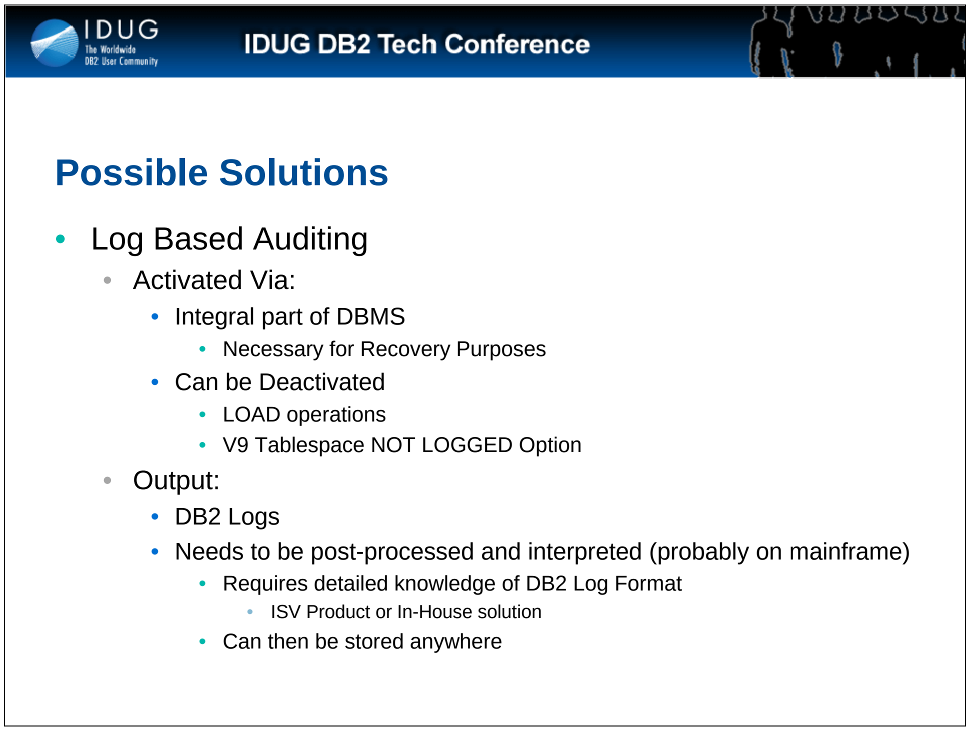

IDUG DB2 Tech Conference

Possible Solutions

- Log Based Auditing
  - Analysis of Information Registered on the DB2 for z/OS Log
    - ***Who*** Accessed a certain Object
      - Authid , Planname, Connection, Unit of Recovery, etc.
    - ***What*** Happened
      - Identifies Object – DBID/OBID + ***RID***
      - Type of Access
        - Update, Insert, Delete
      - Complete Information of the Modification
        - Before Image + After Image
        - With Data Capture Changes can also have the whole ROW
    - ***When*** the access was performed
      - Time of SQL

Log Based Auditing involves the analysis of the DB2 for z/OS Log.

As you are all well aware, the purpose of the Log is to provide short-term ***recovery*** functionality to DB2. However, as the Log contains information about all modified data within the DB2 subsystem, it can provide the canonical Who, What, When of modified data and therefore it may also be useful for Audit purposes....


IDUG The Worldwide DB2 User Community

IDUG DB2 Tech Conference

Possible Solutions

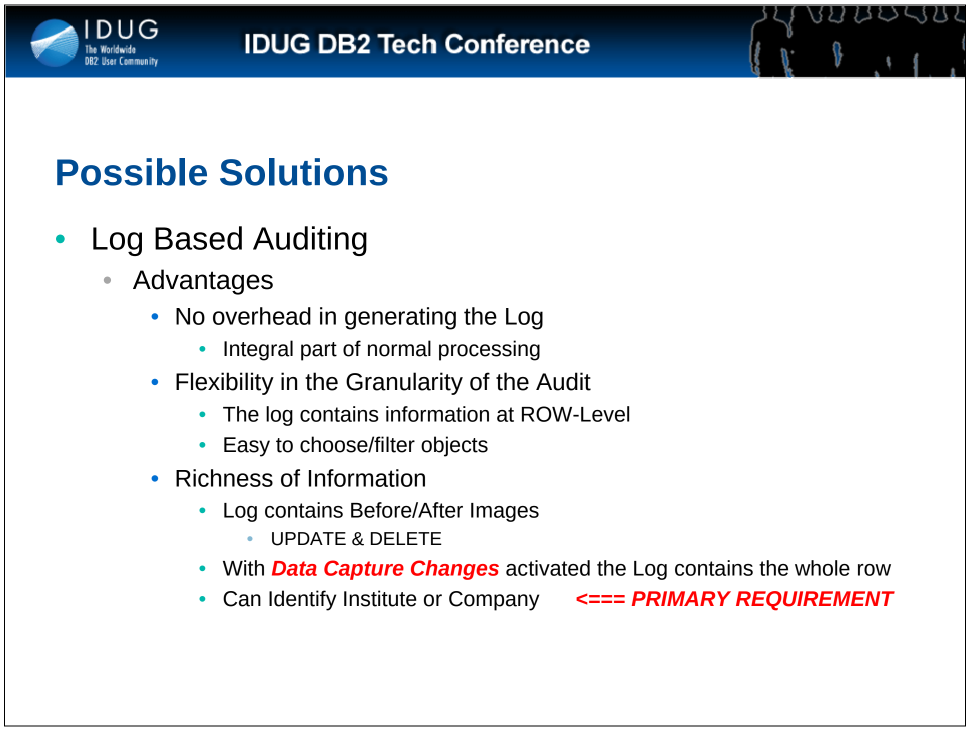
- Log Based Auditing
  - Activated Via:
    - Integral part of DBMS
      - Necessary for Recovery Purposes
    - Can be Deactivated
      - LOAD operations
      - V9 Tablespace NOT LOGGED Option
  - Output:
    - DB2 Logs
    - Needs to be post-processed and interpreted (probably on mainframe)
      - Requires detailed knowledge of DB2 Log Format
        - ISV Product or In-House solution
      - Can then be stored anywhere

Obviously, Log-Based Auditing does not require any activation, however, in certain circumstances it may be deactivated: for example, LOAD LOG NO Utility operations, or from V9 onwards, we may have NOT LOGGED Tablespaces.

The output of Log Based Auditing is obviously the DB2 Logs, and as with the output of Trace Based Auditing, the post-processing of these Logs requires detailed knowledge of the DB2 Log format and therefore either an ISV product is again the really only viable solution.

This initial post-processing will more than likely need to be performed on the mainframe, as the logs are required by DB2. However, the output of this post-processing can then be stored anywhere, and control can be delegated to the Auditors.

Again, as with Trace-based auditing, the requirement of detailed knowledge of the DB2 Log could be an exposure, because whoever post-processes the data could manipulate it.


IDUG The Worldwide DB2 User Community

IDUG DB2 Tech Conference

Possible Solutions

- Log Based Auditing
  - Advantages
    - No overhead in generating the Log
      - Integral part of normal processing
    - Flexibility in the Granularity of the Audit
      - The log contains information at ROW-Level
      - Easy to choose/filter objects
    - Richness of Information
      - Log contains Before/After Images
        - UPDATE & DELETE
      - With ***Data Capture Changes*** activated the Log contains the whole row
      - Can Identify Institute or Company ***<=== PRIMARY REQUIREMENT***

The main advantage of Log Based Auditing is that there is ZERO overhead in generating the audited data as it is automatically gathered as an integral part of normal processing.

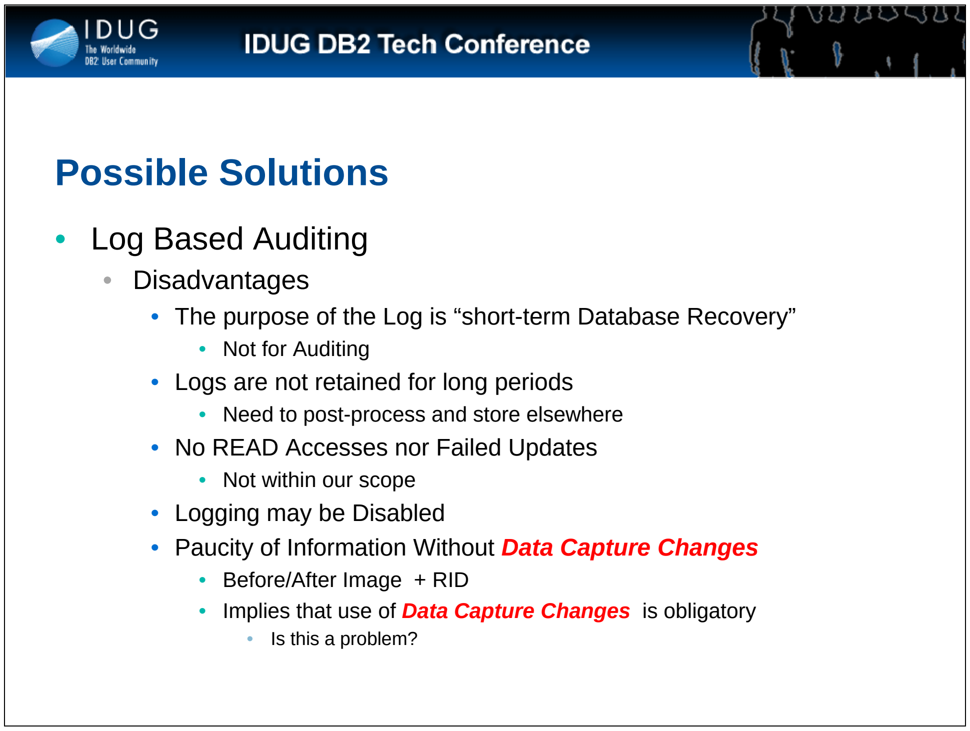
The granularity of the Audit Data is at the ROW level which allows the maximum flexibility on what to Audit.

Also, we have BEFORE/AFTER images of ALL modified data on the Log and, if Change Data Capture is activated, we also have the complete BEFORE/AFTER image of the whole ROW.

In our case we are therefore able to satisfy our PRIMARY REQUIREMENT:

• Identify which Institute has been modified.

Obviously, we also need to understand the table structure to know which column contains the identification of the Institute.



Possible Solutions

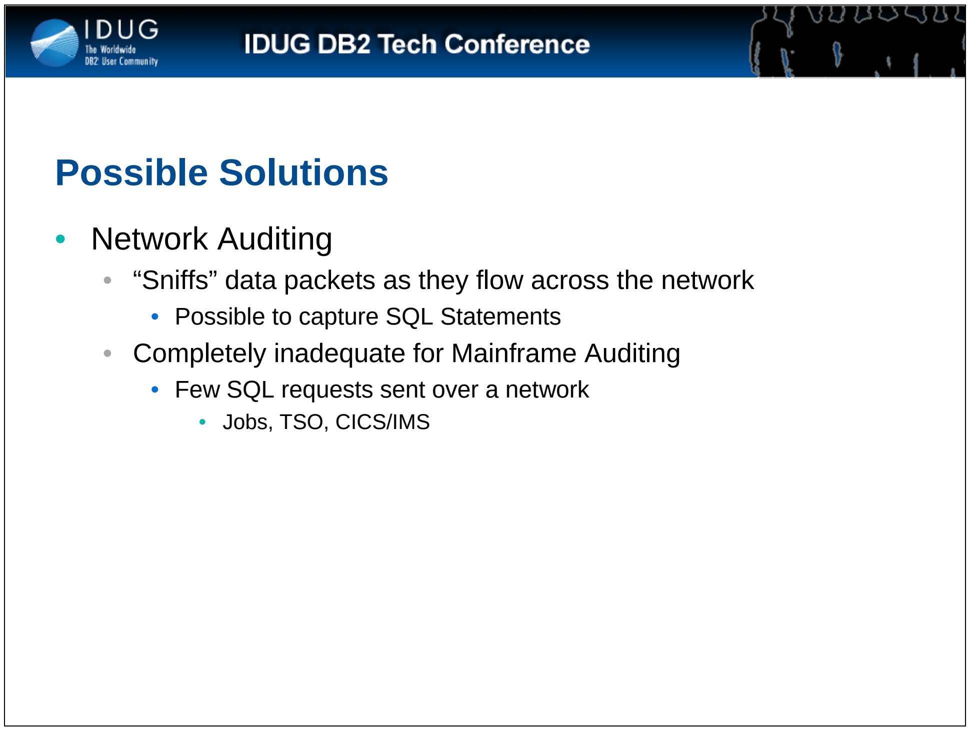
- Log Based Auditing
  - Disadvantages
    - The purpose of the Log is “short-term Database Recovery”
      - Not for Auditing
    - Logs are not retained for long periods
      - Need to post-process and store elsewhere
    - No READ Accesses nor Failed Updates
      - Not within our scope
    - Logging may be Disabled
    - Paucity of Information Without ***Data Capture Changes***
      - Before/After Image + RID
      - Implies that use of ***Data Capture Changes*** is obligatory
        - Is this a problem?

The main disadvantages of Log based Auditing are due to the fact that the purpose of the Log is for “Short-term Database Recovery” and NOT Auditing. This means that the logs are not retained for long periods of time and must be post-processed almost immediately and the output stored elsewhere.

With Log-Based Auditing, obviously neither READ accesses nor FAILED Update attempts are logged...if nothing is modified nothing is written on the Log.

Also, as mentioned earlier, in a few cases LOGGING may be disabled.

Finally, another big disadvantage is the paucity of information provided without activating Data Capture Changes, in fact all we have on the log is the actual value modified together with its previous value and the physical position of the row involved (RID). To get any meaningful information from the Log, Data Capture Changes really is an obligatory choice.


IDUG DB2 Tech Conference

Possible Solutions

- Network Auditing
  - “Sniffs” data packets as they flow across the network
    - Possible to capture SQL Statements
  - Completely inadequate for Mainframe Auditing
    - Few SQL requests sent over a network
      - Jobs, TSO, CICS/IMS

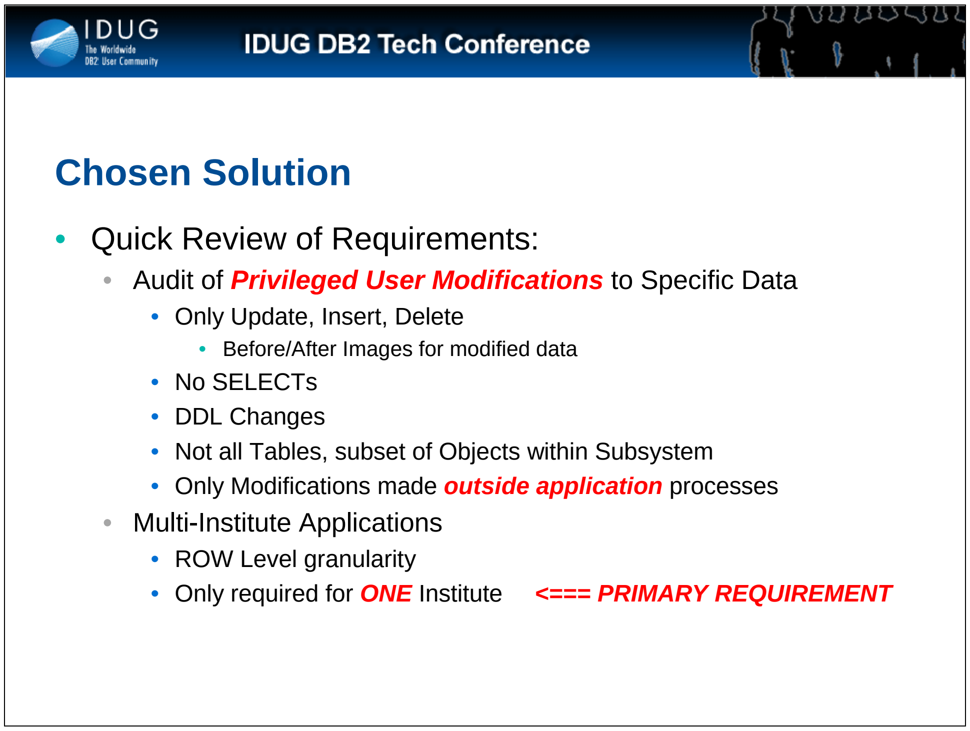
Just for completeness, I will also briefly mention another technique available for auditing even though its use for mainframe applications is practically useless.

This technique is that of “sniffing” data packets for Database accesses as they flow across a network.

Obviously, with DB2 for z/OS, most database accesses originate directly(locally) on the mainframe (TSO, Jobs etc) so there is NO network flow, therefore the use of this technique is severely limited.

Auditing with the DB2 for z/OS Log

- Introduction
- Multi-Institute Applications
- Possible Solutions
- Chosen Solution
- Conclusions

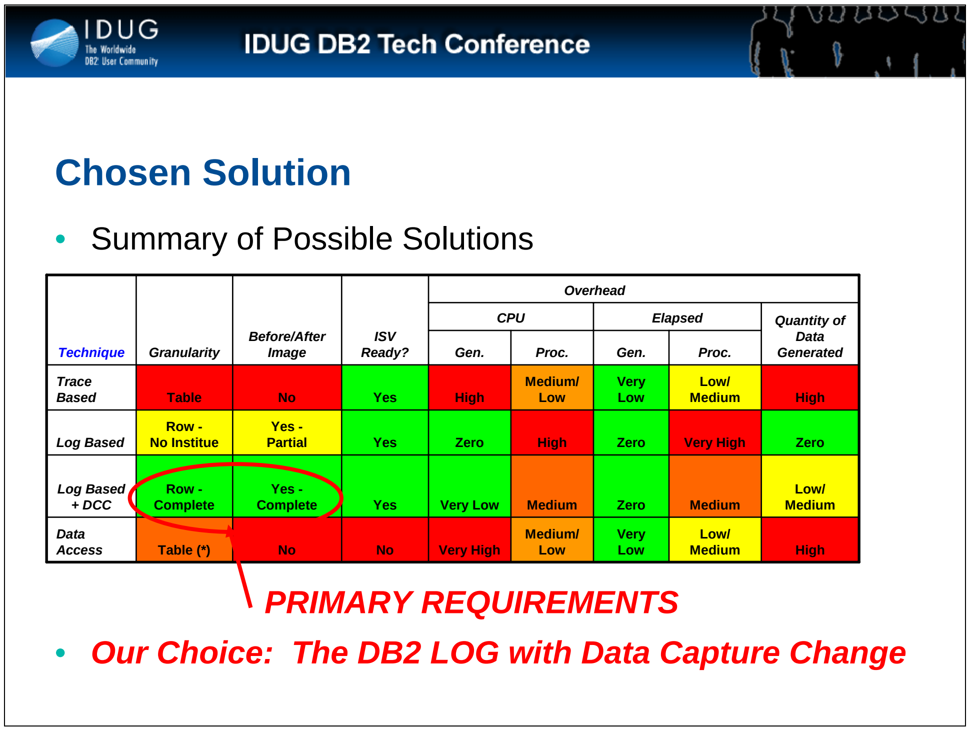

IDUG The Worldwide DB2 User Community

IDUG DB2 Tech Conference

Chosen Solution

- Quick Review of Requirements:
  - Audit of ***Privileged User Modifications*** to Specific Data
    - Only Update, Insert, Delete
      - Before/After Images for modified data
    - No SELECTs
    - DDL Changes
    - Not all Tables, subset of Objects within Subsystem
    - Only Modifications made ***outside application*** processes
  - Multi-Institute Applications
    - ROW Level granularity
    - Only required for ***ONE*** Institute ***<=== PRIMARY REQUIREMENT***

These were our primary requirements... I’m not an auditor so I don’t know whether these are actually necessary...however, that was not our remit.



IDUG DB2 Tech Conference

Chosen Solution

- Summary of Possible Solutions

				Overhead				
				CPU		Elapsed		Quantity of Data Generated
Technique	*Granularity*	*Before/After Image*	*ISV Ready?*	*Gen.*	*Proc.*	*Gen.*	*Proc.*	
Trace Based	Table	No	Yes	High	Medium/ Low	Very Low	Low/ Medium	High
Log Based	Row - No Institue	Yes - Partial	Yes	Zero	High	Zero	Very High	Zero
Log Based + DCC	Row - Complete	Yes - Complete	Yes	Very Low	Medium	Zero	Medium	Low/ Medium
Data Access	Table (*)	No	No	Very High	Medium/ Low	Very Low	Low/ Medium	High

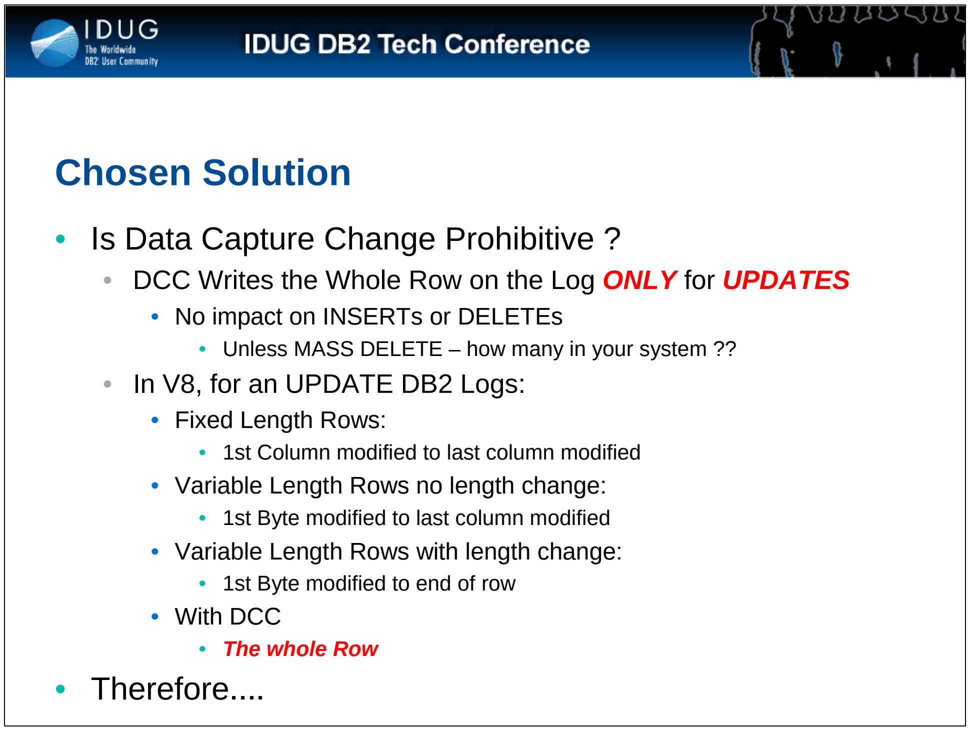
PRIMARY REQUIREMENTS

- ***Our Choice: The DB2 LOG with Data Capture Change***

As can be seen by the chart, the only solution that satisfied fully our primary requirement is that of LOG-BASED Auditing with Data Capture Changes activated.

As we’ve seen previously, the other 2 techniques both had the Show-stopping inadequacy of not being able to satisfy our primary requirements of being able to identify a single institute and to provide complete BEFORE/AFTER images of the data being modified.

However, it must also be noted that we were also aided in our choice by already having in-house an ISV Product which could process and format the DB2 for z/OS Log.
Which will be a god-send when we move to V9 or V10 in that it should be able to handle the Reordered Row Format of the Log.



IDUG DB2 Tech Conference

Chosen Solution

- Is Data Capture Change Prohibitive ?
  - DCC Writes the Whole Row on the Log ***ONLY*** for ***UPDATES***
    - No impact on INSERTs or DELETEs
      - Unless MASS DELETE – how many in your system ??
  - In V8, for an UPDATE DB2 Logs:
    - Fixed Length Rows:
      - 1st Column modified to last column modified
    - Variable Length Rows no length change:
      - 1st Byte modified to last column modified
    - Variable Length Rows with length change:
      - 1st Byte modified to end of row
    - With DCC
      - ***The whole Row***
- Therefore....

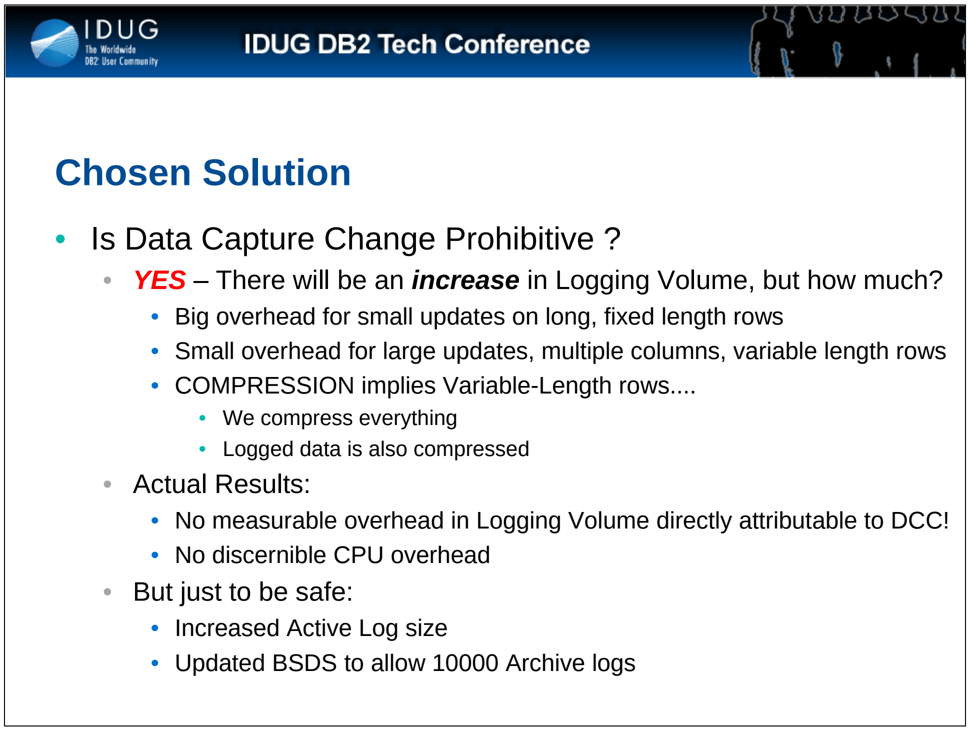
So the first question we had to answer was “Is Data Capture Change Prohibitive?”.

In fact, for some reason, many people are intimidated by Data Capture Change....Probably because with DCC activated, DB2 logs the WHOLE ROW for every single UPDATE and people fear that this will generate enormous volumes of log data.

However, most people overlook the fact that there is absolutely NO Impact on INSERTs or DELETEs!! These continue to be logged in exactly the same way as before.

There is one exception to this however: a MASS DELETE from a segmented Tablespace, which will cause the deletion of every single row and generate an UNDO record on the Log, instead of the normal SPACE MAP update.

So we need to ask the question, how many MASS DELETEs are we performing in production ?


IDUG DB2 Tech Conference

Chosen Solution

- Is Data Capture Change Prohibitive ?
  - ***YES*** – There will be an ***increase*** in Logging Volume, but how much?
    - Big overhead for small updates on long, fixed length rows
    - Small overhead for large updates, multiple columns, variable length rows
    - COMPRESSION implies Variable-Length rows....
      - We compress everything
      - Logged data is also compressed
  - Actual Results:
    - No measurable overhead in Logging Volume directly attributable to DCC!
    - No discernible CPU overhead
  - But just to be safe:
    - Increased Active Log size
    - Updated BSDS to allow 10000 Archive logs

Obviously, with DCC activated, there will be an increase in Logging Volume, but the real questions are:

- HOW MUCH increase?
- Can we Cope with it?

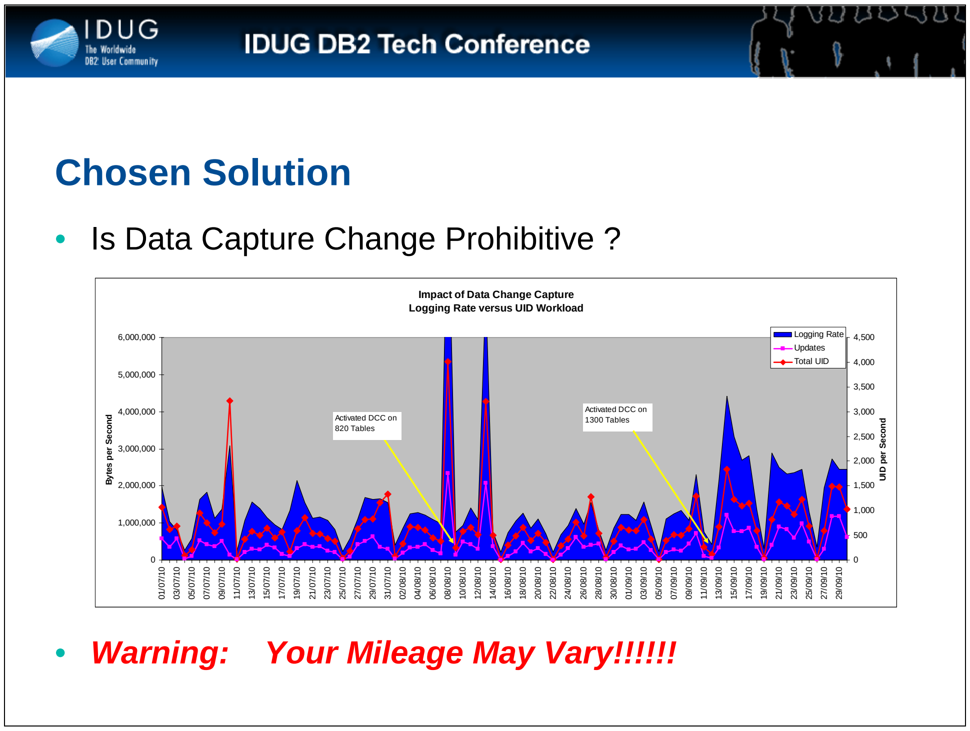
The answer to the first question depends on two main factors:

- The ratio of Updates to Inserts/DELETEs in your subsystem
- The impact of Compression on your data

In our environment, we activated DCC on ALL tables and did not notice any increase in Logging Volume that we could directly attribute to the activation of DCC. Also, we could see NO discernible increase in CPU neither in our applications nor in the MSTR Address Space.

This was an acceptable overhead for us!!

N.B. Your Mileage May Vary!!!!

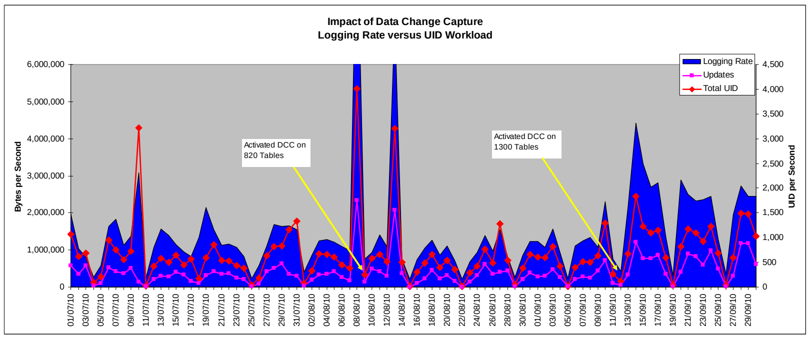


IDUG The Worldwide DB2 User Community

IDUG DB2 Tech Conference

Chosen Solution

- Is Data Capture Change Prohibitive ?



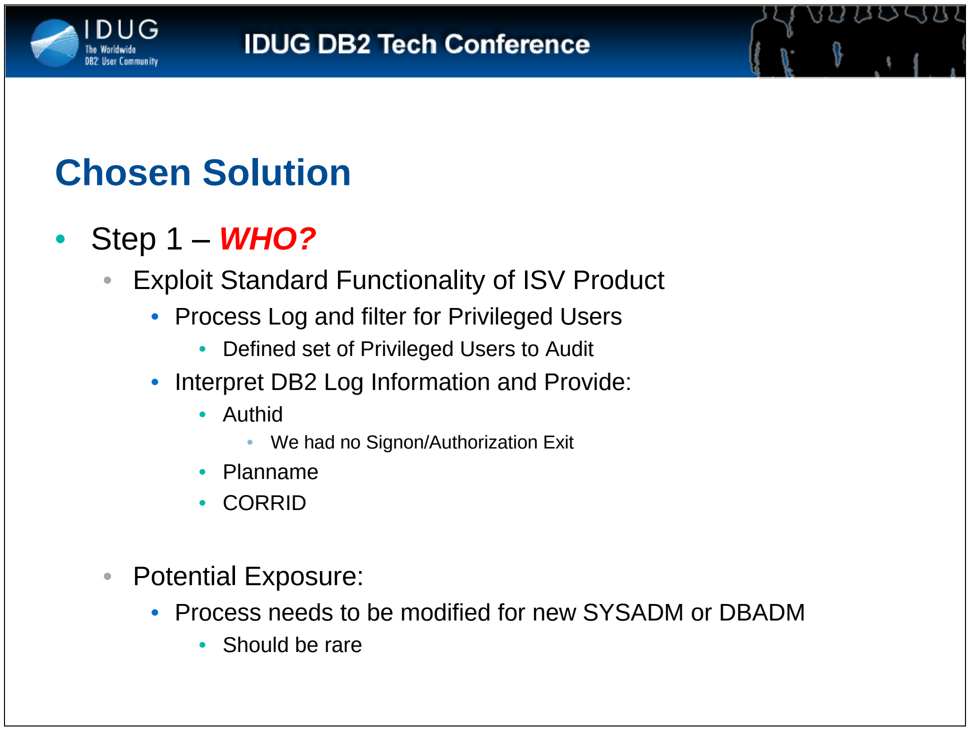
- ***Warning: Your Mileage May Vary!!!!!!***

The above graph shows the average logging rate per day(bytes per second) versus the average number of Updates per second and the number of Updates, Inserts & Deletes per second.

We activated DCC on two groups of the tables, the first group on 820 tables and then 1300 tables in the second group.

As can be seen in the graph the first group had almost zero impact on the logging volume. Whereas the second group coincided, unbeknown to us, with the introduction of new application functionality.

And again, as can be seen from the graph, the increase in logging volume is more related to the change in Application Behaviour rather than the activation of DCC.


IDUG The Worldwide DB2 User Community

IDUG DB2 Tech Conference

Chosen Solution

- Step 1 – ***WHO?***
  - Exploit Standard Functionality of ISV Product
    - Process Log and filter for Privileged Users
      - Defined set of Privileged Users to Audit
    - Interpret DB2 Log Information and Provide:
      - Authid
        - We had no Signon/Authorization Exit
      - Planname
      - CORRID
  - Potential Exposure:
    - Process needs to be modified for new SYSADM or DBADM
      - Should be rare

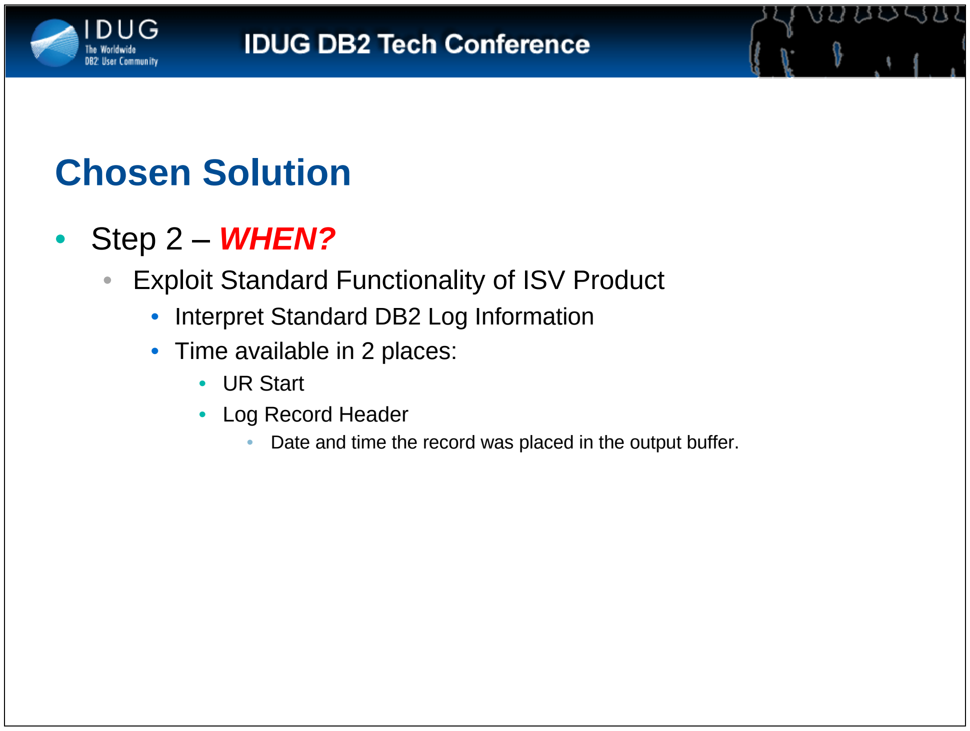
The first step in our process addresses the ***WHO*** issue in auditing.

This was relatively simple in our environment as we had a small list of privileged Users that needed to be audited and we weren't using any Signon or Authorization Exit.

So all we had to do was to use our ISV product to filter the log data using the AUTHID on the Log comparing to our TSO User ID.

The Information provided was AUTHID, PLANNAME and CORRID which was more than enough to satisfy the auditor.

However, our process does have one small exposure and that is, it requires a manual update if the set of Privileged users to audit changes, for example if a new user is given SYSADM authority. However, these should be few and far between.

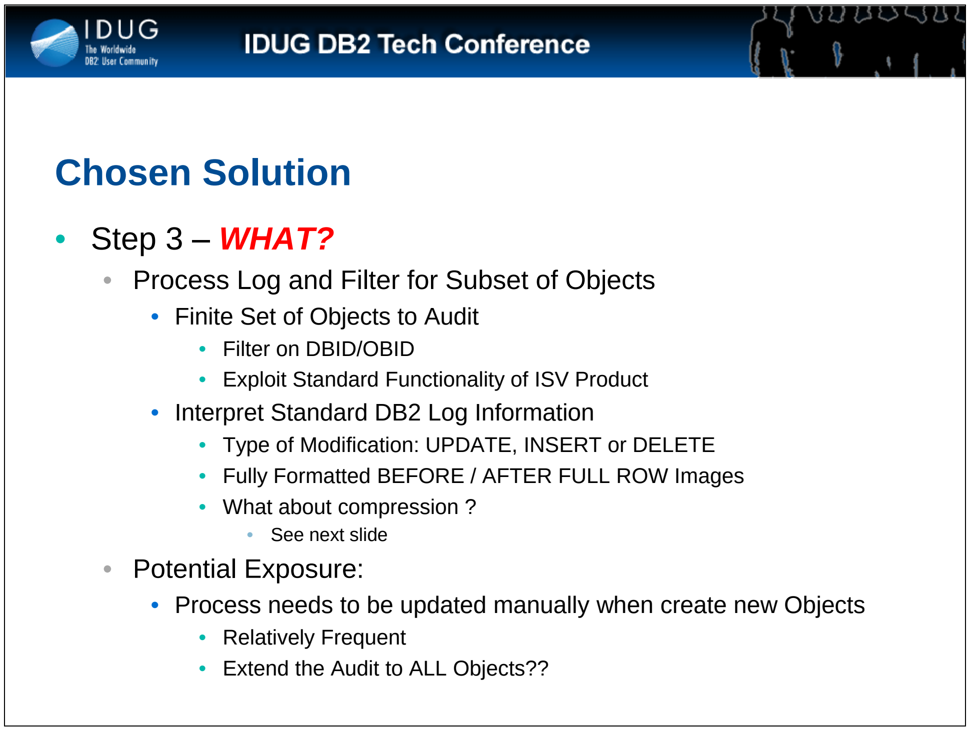

IDUG The Worldwide DB2 User Community

IDUG DB2 Tech Conference

Chosen Solution

- Step 2 – ***WHEN?***
  - Exploit Standard Functionality of ISV Product
    - Interpret Standard DB2 Log Information
    - Time available in 2 places:
      - UR Start
      - Log Record Header
        - Date and time the record was placed in the output buffer.

Satisfying the WHEN requirements of the Audit was extremely simple as we just exploited the standard functionality of the ISV product to interpret the standard Unit of Work information on the DB2 Log.


IDUG DB2 Tech Conference

Chosen Solution

- Step 3 – ***WHAT?***
  - Process Log and Filter for Subset of Objects
    - Finite Set of Objects to Audit
      - Filter on DBID/OBID
      - Exploit Standard Functionality of ISV Product
    - Interpret Standard DB2 Log Information
      - Type of Modification: UPDATE, INSERT or DELETE
      - Fully Formatted BEFORE / AFTER FULL ROW Images
      - What about compression ?
        - See next slide
  - Potential Exposure:
    - Process needs to be updated manually when create new Objects
      - Relatively Frequent
      - Extend the Audit to ALL Objects??

The next step in our process addresses the ***WHAT*** issue of Auditing.

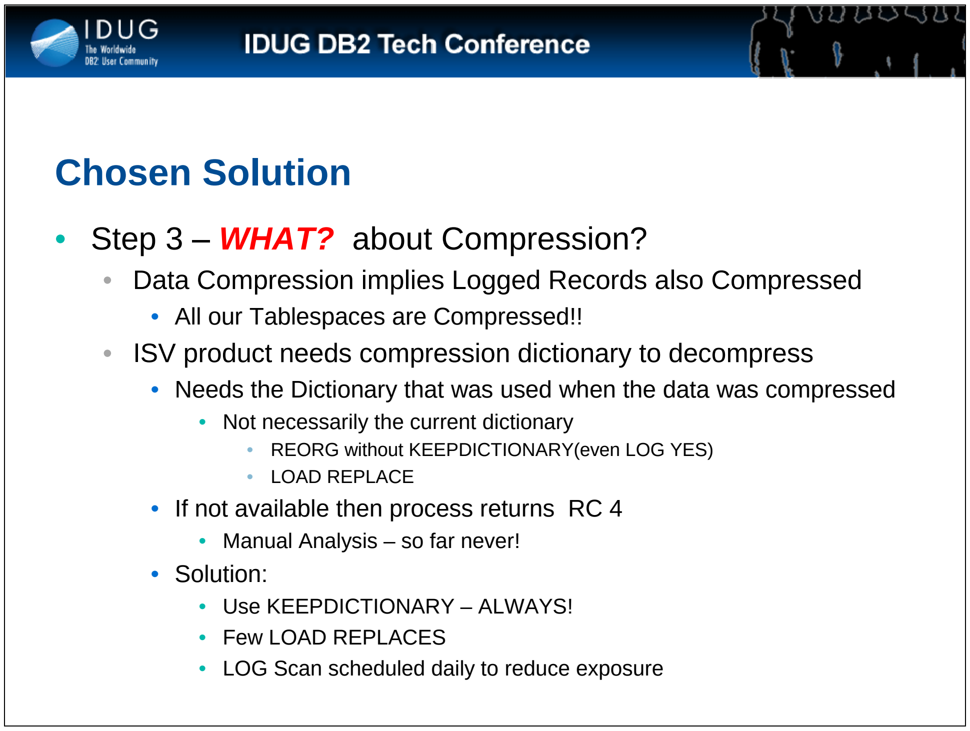
Again, this was a relatively simple process, all we had to do was to use our ISV product to filter the Log data on the DBID/OBID of the objects that we were interested in.   The extracted Before/After Full Row Images were perfectly formatted.

There is however, one small issue with compressed data, which we address on the next slide.

Also, again, we have a small exposure and that is, it requires a manual update if the set of Audit Objects changes.   This is very likely to happen, albeit fairly infrequently.

One option would be to extend the process to all objects within the subsystem, but as we will see later on, this depends heavily on whether it will be necessary to identify the Institute or not.

Another, option is, of course, to move to V10 and use SYSADMs without data access :-)


IDUG DB2 Tech Conference

Chosen Solution

- Step 3 – ***WHAT?*** about Compression?
  - Data Compression implies Logged Records also Compressed
    - All our Tablespaces are Compressed!!
  - ISV product needs compression dictionary to decompress
    - Needs the Dictionary that was used when the data was compressed
      - Not necessarily the current dictionary
        - REORG without KEEPDICTIONARY(even LOG YES)
        - LOAD REPLACE
    - If not available then process returns RC 4
      - Manual Analysis – so far never!
    - Solution:
      - Use KEEPDICTIONARY – ALWAYS!
      - Few LOAD REPLACES
      - LOG Scan scheduled daily to reduce exposure

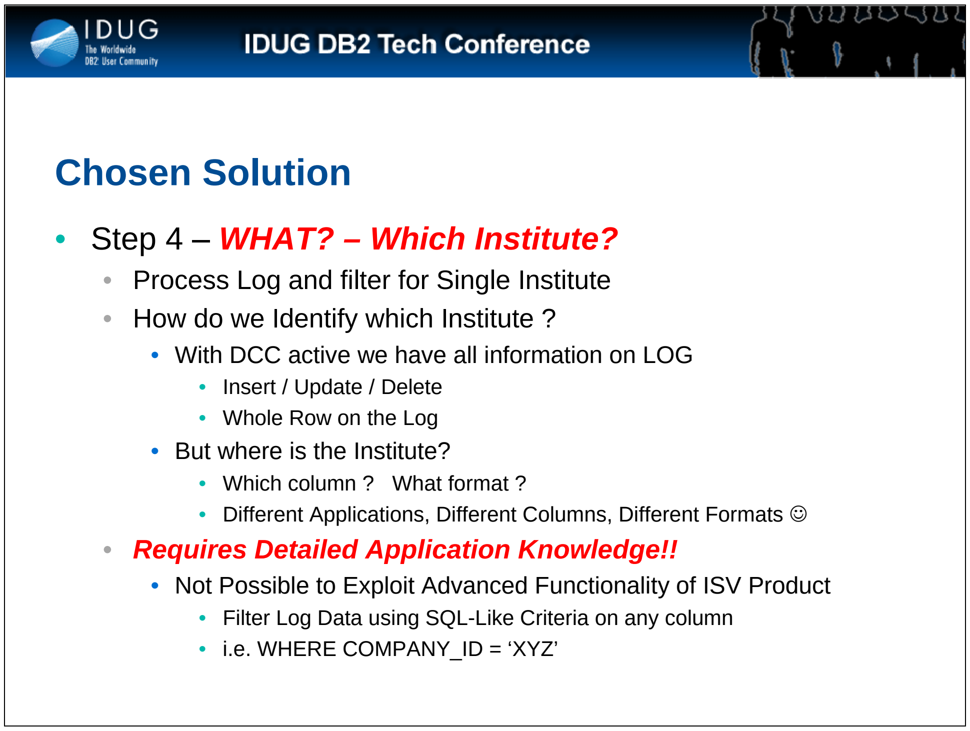
In our environment all of our Tablespaces are compressed. Obviously, this means that all Logged data is compressed, even with DCC. So to decompress the data we need to access the Compress dictionary that was used to compress the data. Remember, the data on the Log doesn't change when we change the Compression Dictionary...

Normally the ISV product that we are using can find the correct dictionary.

However, in a few rare cases, the correct Dictionary is not available, for example after a REORG without Keepdictionary or a LOAD Replace, then in these cases the ISV product will not be able to decompress the data and will end with RC 4.

In this case we need to intervene manually, by re-processing the Logged data and accessing Image Copies to search for the correct Compression Dictionary.

However, in reality, due to few LOAD REPLACEs and standard REORGs always running with KEEPDICTIONARY, the probability of actually hitting this problem is very remote, ...and so far, after one year, it hasn't. Also, to limit our exposure to this problem we are ensuring that we scan the Log every day.


IDUG DB2 Tech Conference

Chosen Solution

- Step 4 – ***WHAT? – Which Institute?***
  - Process Log and filter for Single Institute
  - How do we Identify which Institute ?
    - With DCC active we have all information on LOG
      - Insert / Update / Delete
      - Whole Row on the Log
    - But where is the Institute?
      - Which column ?   What format ?
      - Different Applications, Different Columns, Different Formats ☺
  - ***Requires Detailed Application Knowledge!!***
    - Not Possible to Exploit Advanced Functionality of ISV Product
      - Filter Log Data using SQL-Like Criteria on any column
      - i.e. WHERE COMPANY_ID = 'XYZ'

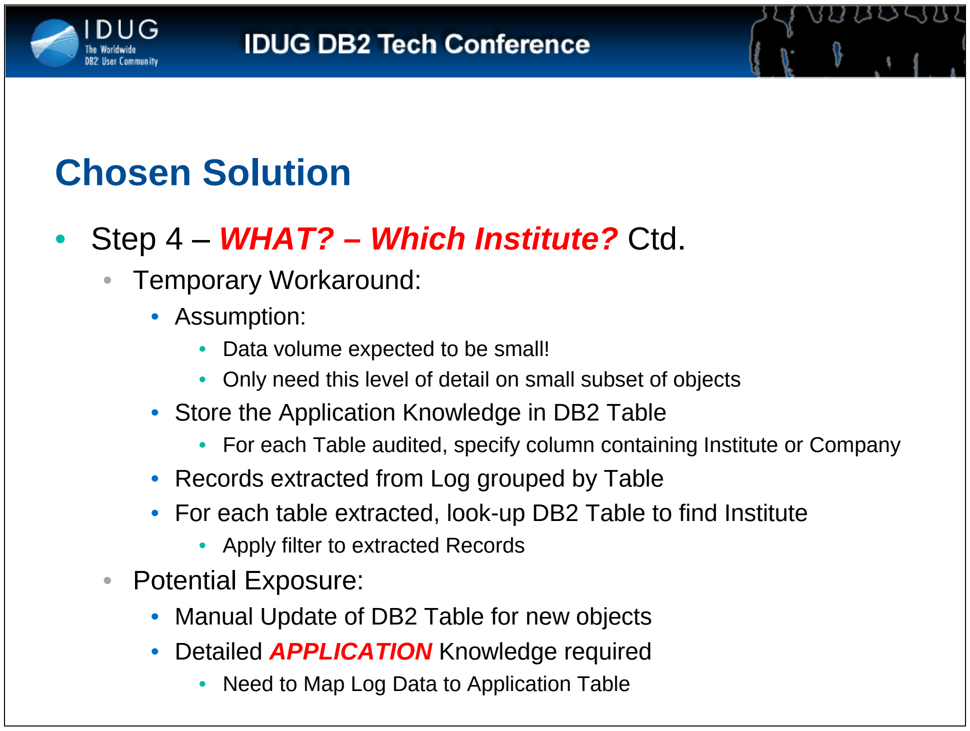
The final part of our process was the most delicate; this was the part where we had to identify the Institute that was being audited.

Due to the use of DCC we had all the necessary information on the Log, however, we were missing one vital piece of information:

• for each table being audited we did not know how to identify the institute.

In fact, the objects being audited were from several applications and each application coded the Institute or Company ID in a different way, normally as the first column of the table, but not always, and each application used differing formats: CHAR(5), DEC(3,0) etc.

Unfortunately, this meant we couldn't exploit the Advanced Functionality of the ISV product which would have allowed us to filter the Log Data using simple SQL-Like criteria on any column in the table.


IDUG The Worldwide DB2 User Community

IDUG DB2 Tech Conference

Chosen Solution

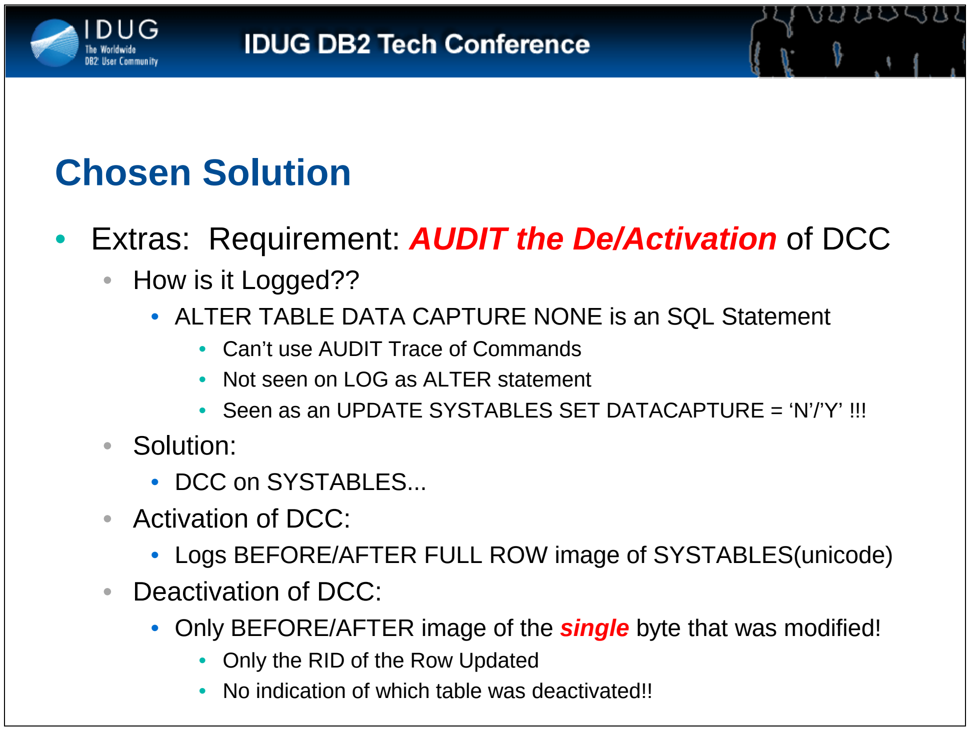
- Step 4 – ***WHAT? – Which Institute?*** Ctd.
  - Temporary Workaround:
    - Assumption:
      - Data volume expected to be small!
      - Only need this level of detail on small subset of objects
    - Store the Application Knowledge in DB2 Table
      - For each Table audited, specify column containing Institute or Company
    - Records extracted from Log grouped by Table
    - For each table extracted, look-up DB2 Table to find Institute
      - Apply filter to extracted Records
  - Potential Exposure:
    - Manual Update of DB2 Table for new objects
    - Detailed ***APPLICATION*** Knowledge required
      - Need to Map Log Data to Application Table

Our temporary workaround to this problem was to store in a DB2 Table the Application Knowledge of which Column contains the Company ID or Institute.

As we process the Log records we look-up this DB2 Table to identify the column containing the company Id or institute and can then apply the correct filter to the extracted Log data.

Obviously this introduces a fairly big exposure to our Audit Process in that we need to update our DB2 table for each new object that will be Audited.

As this information is required for all objects being Audited this will eventually prevent us from Auditing ALL objects to this level of Detail.


IDUG DB2 Tech Conference

Chosen Solution

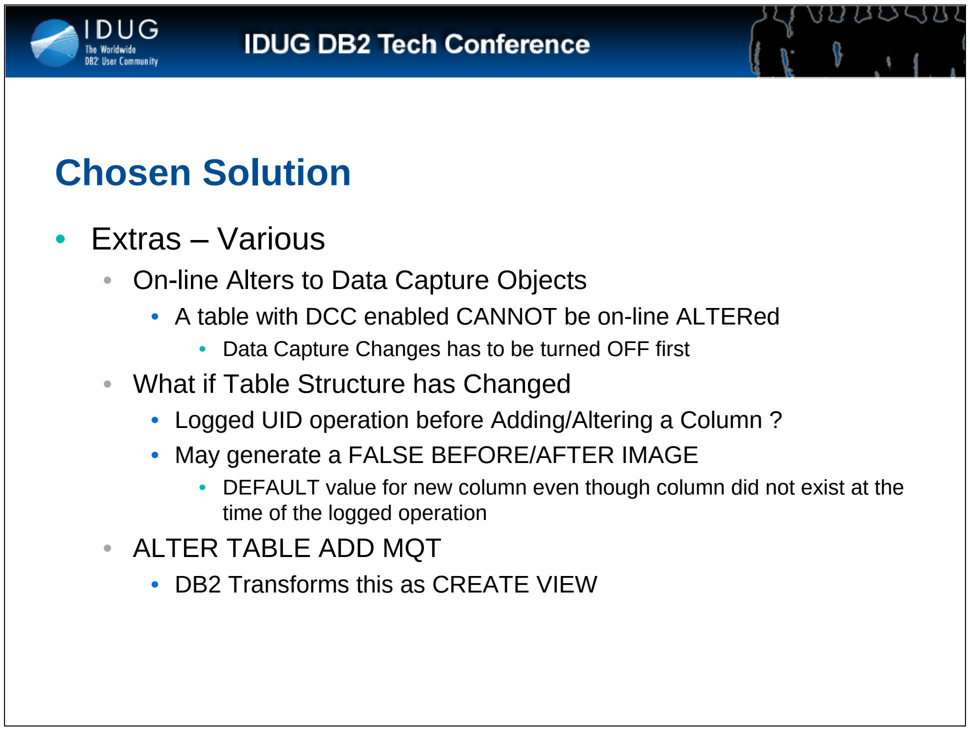
- Extras: Requirement: ***AUDIT the De/Activation*** of DCC
  - How is it Logged??
    - ALTER TABLE DATA CAPTURE NONE is an SQL Statement
      - Can't use AUDIT Trace of Commands
      - Not seen on LOG as ALTER statement
      - Seen as an UPDATE SYSTABLES SET DATACAPTURE = 'N'/'Y' !!!
  - Solution:
    - DCC on SYSTABLES...
  - Activation of DCC:
    - Logs BEFORE/AFTER FULL ROW image of SYSTABLES(unicode)
  - Deactivation of DCC:
    - Only BEFORE/AFTER image of the ***single*** byte that was modified!
      - Only the RID of the Row Updated
      - No indication of which table was deactivated!!

As the use of Data Capture is fundamental to our process, another of our requirements was to AUDIT the Activation and Deactivation of Data Capture on the objects monitored, and this was not as easy as it might seem....

The first problem is that de/activation of Data Capture is done via an SQL Statement, not a DB2 command, so we could not trace it via SMF like other DB2 Commands.

The second problem is that there is NO information on the Log about an ALTER statement!! In fact, it is actually Logged as an UPDATE of the DATACPATURE Column in SYSTABLES!! And, as with all updates, we only get minimal information on the Log, and above all we have absolutely no idea which table was involved.

This implies that DATA CAPTURE is obligatory also for SYSTABLES !!! Also, our ISV Product terminates with RC4 when processing non Data Capture logged rows so we are immediately alerted to any undesired modifications, either to the User Tables or to SYSTABLES.


IDUG DB2 Tech Conference

Chosen Solution

- Extras – Various
  - On-line Alters to Data Capture Objects
    - A table with DCC enabled CANNOT be on-line ALTERed
      - Data Capture Changes has to be turned OFF first
  - What if Table Structure has Changed
    - Logged UID operation before Adding/Altering a Column ?
    - May generate a FALSE BEFORE/AFTER IMAGE
      - DEFAULT value for new column even though column did not exist at the time of the logged operation
  - ALTER TABLE ADD MQT
    - DB2 Transforms this as CREATE VIEW

Just to finish, here are three other facets of Data Capture and the DB2 Log that we came across:

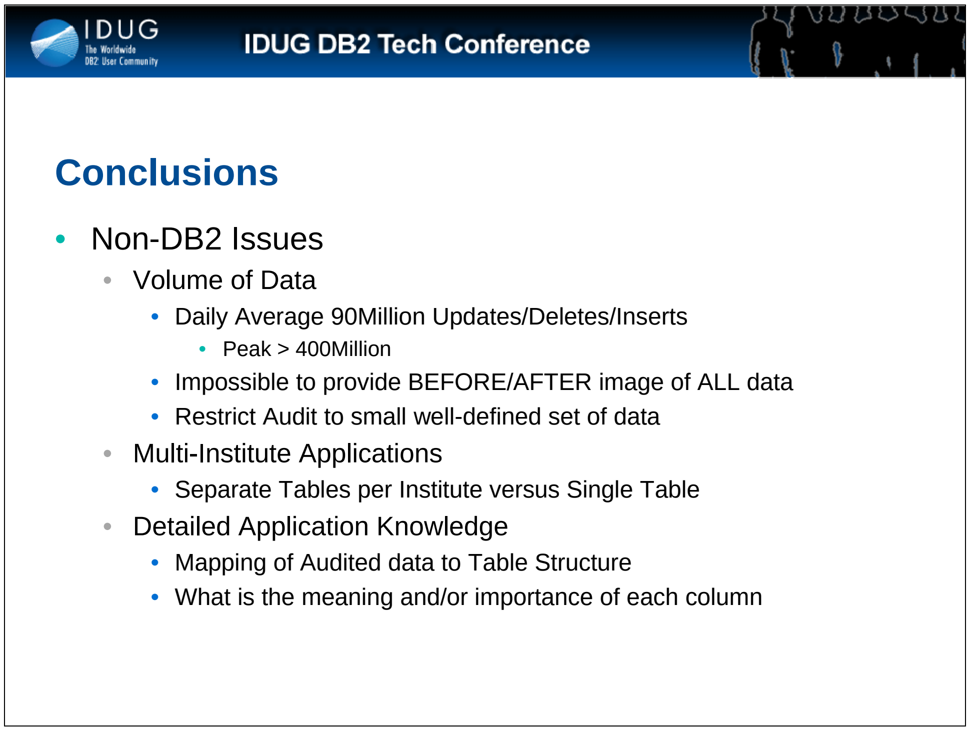
The first is that the use of Data Capture Changes inhibits any On-Line Alters to the objects. So Data Capture must first be turned off before any ALTERs.

The second is that when mapping Logged Data to a table that has subsequently been altered(by deactivating DCC) we may see values for columns that weren't actually present at the time of the actual update. However, this may or may not be a problem depending on your auditor!

And finally, we also noticed that an ALTER TABLE ADD MQT was actually logged as a CREATE VIEW statement....

Auditing with the DB2 for z/OS Log

- Introduction
- Multi-Institute Applications
- Possible Solutions
- Chosen Solution
- Conclusions



IDUG DB2 Tech Conference

Conclusions

- Non-DB2 Issues
  - Volume of Data
    - Daily Average 90Million Updates/Deletes/Inserts
      - Peak > 400Million
    - Impossible to provide BEFORE/AFTER image of ALL data
    - Restrict Audit to small well-defined set of data
  - Multi-Institute Applications
    - Separate Tables per Institute versus Single Table
  - Detailed Application Knowledge
    - Mapping of Audited data to Table Structure
    - What is the meaning and/or importance of each column

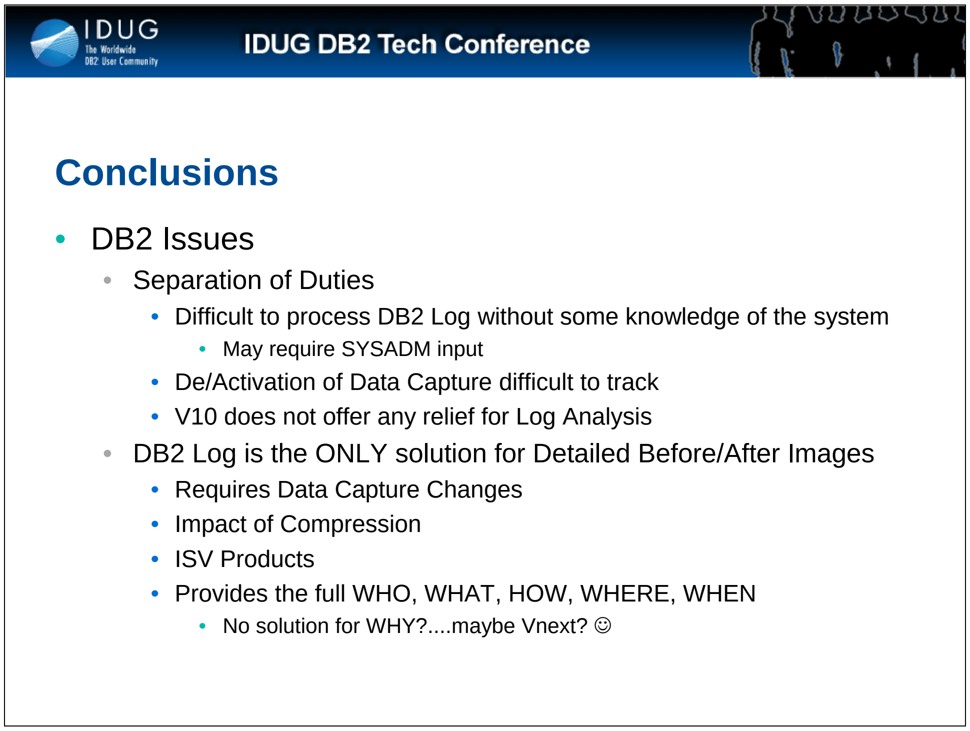
As we have seen, the requirement to provide a Data Modification Audit Trail poses several challenges.

The first of those challenges is the volume of data that may need to be audited.

With a daily average of more than 90 million updates/deletes/inserts it is clearly not practical to audit every single modification but rather it is necessary to restrict the audit to a small well-defined subset of the data.

Secondly, we have seen how the architecture of a multi-institute application can impact the complexity of an audit of a single institute.

Thirdly, any analysis of the audited data will require detailed application knowledge in order to map the audited data to Table structures and to understand the meaning or importance of each modified column.


IDUG The Worldwide DB2 User Community IDUG DB2 Tech Conference

Conclusions

- DB2 Issues
  - Separation of Duties
    - Difficult to process DB2 Log without some knowledge of the system
      - May require SYSADM input
    - De/Activation of Data Capture difficult to track
    - V10 does not offer any relief for Log Analysis
  - DB2 Log is the ONLY solution for Detailed Before/After Images
    - Requires Data Capture Changes
    - Impact of Compression
    - ISV Products
    - Provides the full WHO, WHAT, HOW, WHERE, WHEN
      - No solution for WHY?....maybe Vnext? ☺

The single biggest issue with using the DB2 for z/OS log as the source for the Audit is that of Separation of Duties.

In fact it is extremely difficult to implement a process that analyzes the DB2 Log without the help of the System Administrator. Even the use of an ISV product requires fairly in-depth knowledge not just of DB2 itself but also of the subsystem being analysed. Knowledge typically associated with the DB2 SYSADM and NOT the Auditor!

In this respect the security enhancements introduced in DB2 V10 don’t seem to offer any relief.

However, as we have seen, ,the DB2 for z/OS Log really is the ONLY source for detailed before/after images of modified data. It provides the full WHO, WHAT, HOW, WHERE, WHEN of every single modification....although there is no solution yet for the WHY? :-)

Thank You!!

IDUG The Worldwide DB2 User Community

DB2 Tech Conference | Prague, Czech Republic

Adrian Collett
Expertise4IT s.r.l.
adrian@expertise4IT.com

Session: E1

Who, What, Why, Where & When?
Auditing with the DB2 for z/OS Log