

3rd Worldwide Firebird Conference 2005, Prague, Czech Republic

Audit Trails, Transaction Log, Redo with the IB LogManager product family

Session: TOOLS-P208-R

Thomas Steinmaurer DI

Upscene Productions

Database Tools for Developers

Database Workbench, LogManager Series, Advanced Data Generator,
InterXpress for Firebird

<http://www.upscene.com>



Firebird-Conference
Prague 2005

Overview

➡ § Motivation

§ Application areas

§ Server-side audit trail

§ IB LogManager product family



Motivation

- § Firebird's Multi-Generational Architecture (MGA) – A flexible approach for concurrency control
 - Firebird doesn't use page/table/record locks to ensure a consistent view of data
 - Firebird maintains different versions of one record („Versioning“)
 - Each change of a record is reflected in a new record version
 - Back (old) record versions only for internal (engine) use
 - „Readers don't block writers and writers don't block readers“

- § Where is a transaction log?
 - Due to the MGA, Firebird doesn't have/need a transaction log file
 - Recovery is simply rolling active transactions back

○ Several reasons to keep track of data changes



Overview

§ Motivation

➔ § Application areas

§ Server-side audit trail

§ IB LogManager product family



Application areas (1/2)

- § Log any kind of data changes
- § To answer questions like: „When, and by whom, were certain records modified, deleted or inserted?“
- § Logs for statistical purposes (price fluctuations, ...)
- § Tweak your client application concerning DML operations
- § „Black Box Debugging“



Application areas (2/2)

- § Audit trails as a necessity to conform to different application domain quality standards (US Food & Drug Administration, ...)
- § Change log for asynchronous replication
- § DML notification mechanism (e.g. IBO Global DML Caching)
- § And probably much more ...



Overview

§ Motivation

§ Application areas

➡ § Server-side audit trail

§ IB LogManager product family



Server-side audit trail (1/5)

§ Client vs. Server-side logging mechanism

- Client-side: Track data changes at the client
- Server-side: Track data changes at the server

§ Design goal

- Client-independent
- Best possible performance => less network traffic
- => Server-side audit trail is the way to go!

§ Firebird offers everything we need for a server-side audit trail

- Metadata (tables/fields, stored procedures, ...)
- Triggers (server-side code module that gets executed automatically in response to a data manipulation operation)



Server-side audit trail (2/5)

§ Metadata

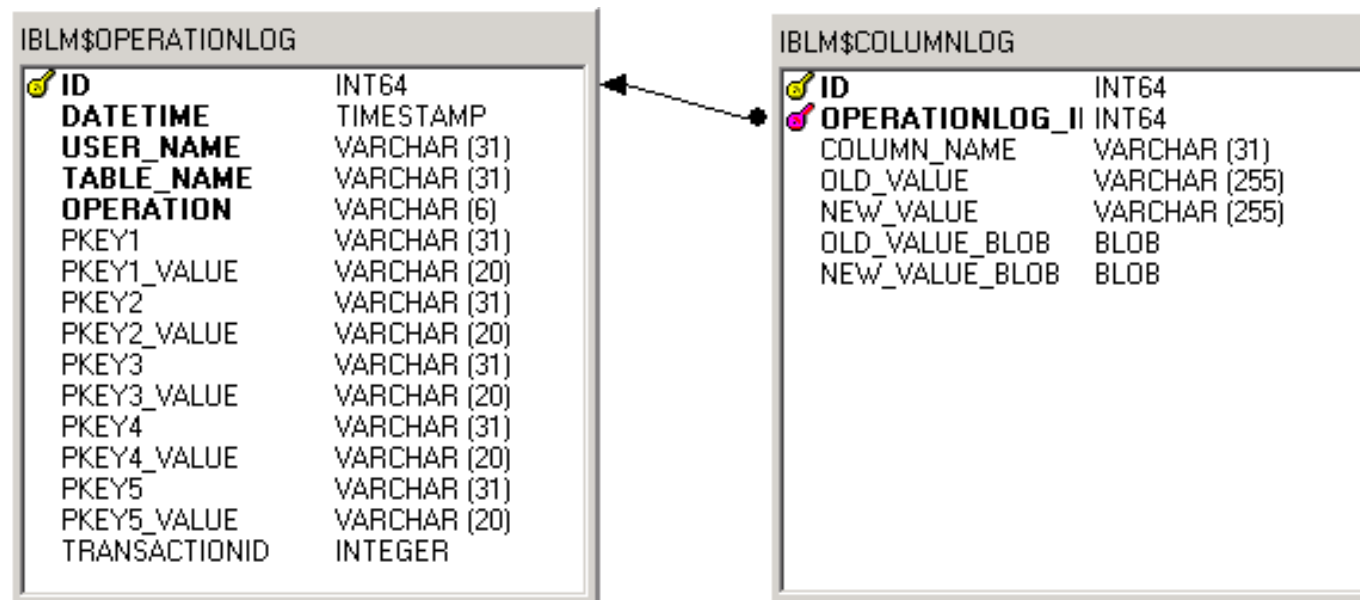
- Approach 1: A log table for a particular business requirement (e.g. tracking article price changes)
- Approach 2: One log table for each tracked table (doubles the number of tables and log table needs to be in-sync with regular table regarding DDL)
- Approach 3: One universal log table, storing log data for each tracked table

§ Approach 3 is subject of this session, also used by the IB LogManager product family



Server-side audit trail (3/5)

§ Log tables for approach 3



Server-side audit trail (4/5)

§ Trigger example for approach 3 – Track UPDATE operations

```
CREATE TRIGGER TRI_ROOM_L_D FOR ROOM
ACTIVE AFTER UPDATE POSITION 32767
AS
DECLARE VARIABLE VAR_ID NUMERIC(18,0);
BEGIN
  IF (USER <> 'IBLM_REPL') THEN BEGIN /* Conditional logging => optional */
    EXECUTE PROCEDURE SP_INSERTINTOIBLM$OPERATIONLOG (
      'ROOM', 'UPDATE', 'ID', OLD.ID, ..., CURRENT_TRANSACTION)
    RETURNING_VALUES :VAR_ID;
    IF (((OLD.PICTURE IS NULL) AND (NEW.PICTURE IS NOT NULL)) OR
        ((OLD.PICTURE IS NOT NULL) AND (NEW.PICTURE IS NULL)) OR
        (F_IBLM_BLOB_CRC(OLD.PICTURE) <> F_IBLM_BLOB_CRC(NEW.PICTURE))) THEN
      INSERT INTO IBLM$COLUMNLOG (
        ID, OPERATIONLOG_ID, COLUMN_NAME, OLD_VALUE,
        NEW_VALUE, OLD_VALUE_BLOB, NEW_VALUE_BLOB)
      VALUES (GEN_ID(GEN_IBLM$COLUMNLOG_ID, 1), :VAR_ID, 'PICTURE', NULL,
        NULL, OLD.PICTURE, NEW.PICTURE);
    ...
  END
END
```

Special treatment of NULLable fields

Operation Log

Column Log

UDF needed to compare BLOB data



Server-side audit trail (5/5)

- § Concept of building a server-side logging mechanism is easy
- § Creating metadata isn't that much work, but
- § Creating and maintaining triggers manually is annoying and error-prone
- § The IB LogManager product family is a real time-saver here



Overview

§ Motivation

§ Application areas

§ Server-side audit trail

➔ § IB LogManager product family



IB LogManager product family (1/4)

- § IB LogManager: Core product to integrate and maintain a server-side auditing mechanism
- § IBLMViewer: For browsing, exporting and printing log data, without the capability to change the logging schema
- § IBLMExport: Command-line utility to export log data into different file formats
- § IBLMPump: Command-line utility to move log data from the production into a „transaction log“ database
- § IBLMRedo: Wizard-driven utility to re-apply tracked changes on a destination database



IB LogManager product family (2/4)

§ Key features

- Supports InterBase 5.x, 6.x, 7.x and Firebird 1.x, 2.x
- Server-side logging mechanism
- Table operation logging
- Table column logging (including BLOBs)
- User-definable conditional logging
- Scripting Engine to automate repetitive tasks
- Browse, export, print log data
- Multilingual graphical user interface
- Global IBO DML Caching Integrator Support
- Several addons (IBLMViewer, IBLMExport, IBLMPump, IBLMRedo)



IB LogManager product family (3/4)

§ Pricing

- IB LogManager
 - 1-4 license(s): 69 EUR / license
 - 5-9 licenses: 56 EUR / license
 - 10 or more licenses: 51 EUR / license
 - Site license: 399 EUR / license
- Addons (for each)
 - 1-4 license(s): 29 EUR / license
 - 5-9 licenses: 24 EUR / license
 - 10 or more licenses: 22 EUR / license
 - Site license: 169 EUR / license
- Productivity suite: Database Workbench Pro for Firebird and IB LogManager product family
 - License price: 179 EUR

§ More information: <http://www.upscene.com>



IB LogManager product family (4/4)

Demo

