FM AuditLog Pro 2.0

Product description

FM AuditLog Pro 2.0 is, as its name implies, a data audit trail. It records every creation/modification in a table from where you can trace data activity and roll back.

Unlike many audit log techniques, it does not store the log on the modified record itself but in a separate table, or even a separate file (which we recommend because this table will tend to become quite big, and also because if your main file has a problem, at least the log is safe)

Requirements

FileMaker Pro 14 required

FileMaker Server 14 optional (if your database is hosted on a FileMaker Server 14 allowing Script sessions, then the heavy work will be done server side, without any configuration)

What does it log?

Every data modification, table per table (you might want to choose not to audit some tables). FM AuditLog Pro 2.0 logs every modification in standard fields (not calculations, globals or summary fields).

Standard fields include Text, Number, Date, Time, Timestamp, and Container types, including repeating fields.

Important: FM AuditLog Pro does NOT log record deletions. That's because FileMaker does not provide tools to know which records will be deleted via relationships. We chose to do nothing rather than do something that would work only in some situations. It means that if you want to log deletions, you have to script them and write your own log. However, the roll-back feature is able to roll back a record that has been deleted.

When does it log?

Keep in mind that the log is evaluated by the calculation engine in an auto-enter global field, which means that you have nothing to do once the required fields are copied to your solution tables.

BUT, very important: the log is written to the log table by script, which means that this script has to be triggered somehow. We recommend that you add an onRecordCommit script trigger on each of your layouts.

There are some specific cases (Import, Replace Field Contents, Drag and Drop from an external application), but we'll see this later in this document

Installation

1 - Open the file 1MT_FMAuditLogPro2.fmp12

2 - From the left menu, choose re-login and use the account name/password you received with your licence.

3 - Import the ΩAUDITLOG table from the demo file to your solution or to a new file. **We recommend creating a new file. If you do so, remember that it will be a data file only. All scripts and layouts will be in your solution file.**

3b - If you chose to create a new file, add a file reference to this new file in your solution file.

>> you can now copy the script "ΩAUDITLOG | setup report" to your solution file, which will help you in detecting what is missing in the file. The script will generate an HTML file "AUDITLOG_InstallationReport.html" on your desktop. <<

4 - Ensure there is a table occurrence and a layout named ΩAUDITLOG in your solution file and that they are well representing the ΩAUDITLOG table

5 - add a second ΩAUDITLOG table occurrence to the graph and name it ΩAUDITLOG_getOldValue **(copy paste the name, it has to be exactly the same during the install process)**

5b - create a relationship like:

	ΩAUDITLOG		ΩAUDITLOG_getOldValue
	combinedKey	=	combinedKey
AND	serial	≠	serial

6 - add an ΩAUDITLOG table occurrence to the graph and name it ΩAUDITLOG_otherTransactions **(copy paste the name, it has to be exactly the same)**

6b - create a relationship like:

	ΩAUDITLOG		ΩAUDITLOG_otherTransactions
	ID_transaction	≠	ID_transaction
AND	ID_record	=	ID_record

7 - add an ΩAUDITLOG table occurrence to the graph and name it ΩAUDITLOG_endOfTransaction **(copy paste the name, it has to be exactly the same)**

7b - create a relationship like:

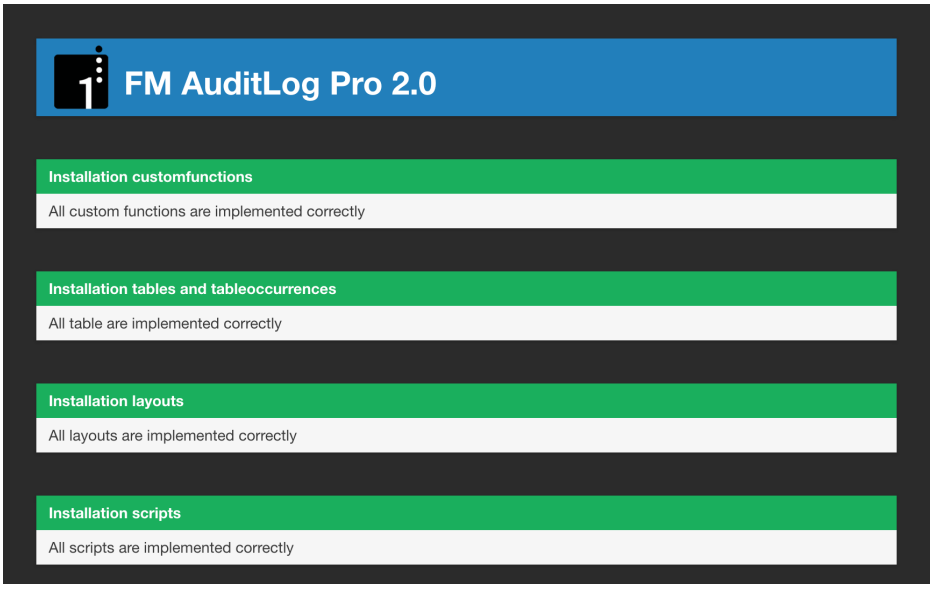
	ΩAUDITLOG		ΩAUDITLOG_endOfTransaction
	ID_transaction	=	ID_transaction
AND	one	=	isEndOfTransaction

8 - Import the custom function AUDITLOG from the demo file to your solution file(s). (FileMaker Pro Advanced required)

- AUDITLOG

9 - Import the script folder "ΩAUDITLOG <<< COPY TO YOUR SOLUTION" (you can rename this group)

10 - Run the script “ΩAUDITLOG | setup report” to test if all necessary tables, table occurrences and layouts are installed. The upper part of the report should look like this. If you see any error, correct them before proceeding.



Implementation

Now you have everything. Let's implement to your tables.

1 - In the demo file, go to the field definition of the Contacts table (Manage Database)

2 - Select the 2 fields starting with ΩAUDITLOG and copy them (FileMaker Pro Advanced required)

- ΩAUDITLOG
- ΩAUDITLOG_pk

3 - Paste these fields to the solution tables that you want to audit

3 a) if implementing to a table with existing records, make sure ΩAUDITLOG_pk contains values. Tip: make the ΩAUDITLOG_pk a calculated field (type text) and close the Manage Database window. Then change the field back to an auto enter text field (uncheck the “do not replace existing field value (if any)”). This way the field will contain the calculated values and the modification date won’t be updated.

4 - And now the painful operation: go to each and every layout of your solution, and add these script triggers (browse mode only)

- trigger OnRecordCommit, script: "ΩAUDITLOG | onRecordCommit"
- trigger OnRecordRevert, script: "ΩAUDITLOG | onRecordRevert"

If you already have set those scripts triggers to perform other scripts, then modify those scripts so they Perform the AuditLog scripts.

5 - (optional). For each base table of your solution, create a layout named

ΩAUDITLOG_RB_[baseTableName]

This layout will be used to roll back faster. (RB stands for Roll Back)

A baseTableName is what you find under the Manage Database/Table panel, not the table occurrences of your relationship graph.

A good way to find all the base table names is to paste this calculation to your Data Viewer:

```
ExecuteSQL ( "SELECT DISTINCT baseTableName FROM FileMaker_Tables" ; "" ; "" )
```

Important: if you don't create these layouts, ensure that:

- you have at least 1 layout for each base table
- this layout does not have a script trigger onLayoutEnter or onLayoutLoad that would redirect the user to another layout or modify the data (typically, recurring

imports can be a problem). If you have one of these, please consider creating the ΩAUDITLOG_RB_[baseTableName] layouts

You're set!

The above steps do not reproduce the interface we included in the demo file. It might be relevant or not. You're free to design your own interface, and we even encourage this (the one in the demo file might be good to understand the principle, but certainly not perfect in a real world solution)

Understanding the fields of the ΩAUDITLOG table

In the log table, most fields are self explanatory (and commented), but let's take a look to some fields that are going to be very important to use/display the log in your solution.

ID — this is the table primary key. To roll back a single value, you should send this field as a parameter to the roll back script.

ID_record — this is your record UID, found in the ΩAUDITLOG_pk field. We have to use this system because not everyone uses UIDs as primary keys (not everyone uses primary keys by the way). Now if you already use UIDs, you could modify the system in order to avoid having a second UID.

To display the log for a given record in your interface, create a relationship from ΩAUDITLOG_pk (in your table) to ID_record in a new ΩAUDITLOG table table occurrence, and use a portal.

You might want to roll back a record or a set of records to a timestamp. The parameter to send to the Roll-back script is then the timestamp you want to revert to, then a ¶, then a ¶ delimited list of record UIDs. Example:

```
List ( $timestamp ; $record_uid.list )
```

ID_transaction — this allows you to spot all the modifications that were committed together, as a single transaction.
Important: a transaction in our system is defined by a time the AUDITLOG sript is run. This is why it is important to trigger it on each

	commit (onRecordCommit). As you will see in "Special situations", an import will be considered as a single transaction. You might want to revert one or more transactions. In this instance, send a ¶ delimited list of ID_transaction as a parameter to the roll-back script.
isNewRecord	boolean. 1 if the record is new (was never logged before). **Important**: reverting a transaction where a record was created will delete this record (and potentially its related record if you defined some relationships to delete related records)
serial	simple serial number. We need this in the relationships you created earlier. Take a look at the "Roll back to timestamp" example as well from the portal/"more" popover in the demo file.
foundCount	we need this one in our scripts, but you can also use it to display the number of related ΩAUDITLOG faster than using Count().
listOfIDs	a summary field to list the current found set IDs. It allows reverting a custom set of modification from the ΩAuditLog context or thru a relationship (in the demo file, you'll find an example on the table view (Roll forward found set)
ΩAUDITLOG_result	is a global field where you can read the result of the log and roll-back scripts. You can choose to expose it to the user.

Special situations

There are 4 different situations that you should consider closely.

The three first are related to what FileMaker calls "auto-commit", which means a record can be modified without being open, thus causing the onRecordCommit script to be inefficient.

These 3 situations have to be handled differently. Here is a memo (there are also examples in the demo file)

1 - Imports. There are 2 things to remember about imports:

- the perform auto-enters checkbox must be enabled during imports
- the script "ΩAUDITLOG | onRecordCommit" must be performed explicitly after the import. (won't be triggered by the layout script trigger)

2 - Replace fields contents. You can replace field contents from a committed record. The user can do it by just placing the caret in a field, not modifying it, and select Replace, and a script can do it even more frequently. To ensure the log will trace the operation as a single transaction, be sure to open the record before replacing. You can use the Open Record script step.

3 - Drag and drop from an external application or from another FileMaker window. If you drag some content onto a field of a committed record, the record will be modified, but not opened. (this is not true if you drag content from another field or a web viewer of the same window). Therefore, what you need to do is to add a script trigger onObjectModify to objects where the user might want to drag-drop (typically, container fields), and have the triggered script open the record. The demo file illustrates this with the container object.

4- The 4th situation you might have to deal with is when you're working with Custom Web Publishing, SQL Plug In, or ODBC. In these situations, like for Imports, you'll have to explicitly run the script "ΩAUDITLOG | onRecordCommit" after an update/ insert.

Using the rollback

The rollback provides a mechanism to restore the state of a record to the state just before a certain modification, transaction or timestamp.

The ROLLBACK script can perform a rollback based on an ID (of the log table), which will revert that specific modification, based on an ID_Transaction, which will revert the complete transaction, or based on a timestamp and an ID_Record that will restore the record to the state just before that timestamp.

For faster performance it is better to create a layout named “ΩAUDITLOG_RB_<<basetableName>>”. If this layout is not found, the rollback will be performed on the first layout that is based on a correct table occurrence. However this can abort the action if for example script triggers (on that layout) would navigate to another context or another record.

How to use the “ΩAUDITLOG | ROLL BACK” script

Revert field modifications

Parameters:
- list of UUIDs (ID)

The script will find all log records for the specified IDs, and will revert each field modification (restores the valueOld field).

It is important to note that if a list of modifications for the same field is provided, the last modification will be restored first, moving up the time line, and the last modification will be restored as the latest.

Revert complete transactions

Parameters:
- list of UUIDs (ID_Transaction)

The script will find all log records for the specified transaction IDs, and will revert each field modification (restores the valueOld field).

It is important to note that if multiple transactions for the same record are provided, the last modification will be restored first, moving up the timeline, and the last modification will be restored as latest.

Perform a rollback to a certain timestamp

Parameter:
- 1st line: timestamp
- list of UUIDs (ID_Record)

The script will find all log records for the specified record ID after a certain timestamp, and will revert each field modification.

It is important to note that the script will move backwards from now to the timestamp provided, restoring each modification.

Note on record re-creation

If a record which is part of a rolled back transaction was deleted in the meantime, it will be recreated with all available information in the audit log.
Please note that if several records are recreated during a single transaction, they might be created in a different order than the original order. (before the deletion)

Custom Delete script

During a roll back, it could be that some records have to be deleted (if you roll back a transaction during which some records were created).

Before deleting a record, a subscript will be performed: ΩAUDITLOG | Delete | Custom.

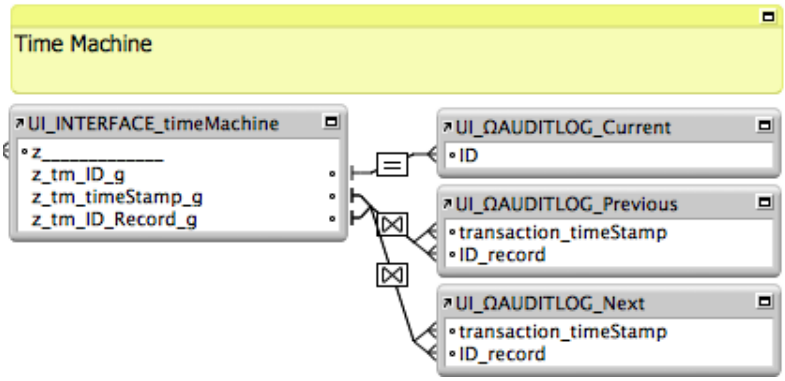
You can customize it to keep a trace of deleted records.

Important: the rollback will only perform the deletion if the script returns true (1): Exit Script [True] or Exit Script [1].

Implementing the Time Machine

If you want to visualise the lifetime of a record and see which fields where modified at a given time, you can implement a 'Time Machine' like interface.

The Time Machine consists of 1 script and 4 table occurrences.



To reproduce the UI, we created a simplified version on a hidden layout named "time machine explained". This version exposes tricks we used to make it nice.

License terms

Like you, we hate reading license terms. So we've made them very short. Please read them carefully and respect them:

Your purchase of a FM AuditLog Pro 2.0 license allows you to use the product as much as you want in as many solutions as you want for as many users as you want.

The only thing you can't do with your license is give it or sell it to someone else.

For instance, if you are a professional FileMaker developer/consultant, use it for your customers' solutions, but please do not share with another consultant. Instead, ask him to buy a license at www.1-more-thing.com