

Under the Hood: Draco Database Engine

COR002

Clay Maeckel

Chief Software Architect at FileMaker, Inc.

© 2016 FileMaker, Inc.



Hello my name is... Chief Software... reporting to Frank Lu, VP of Engineering

Who is Clay Maeckel?

Been working at Apple/Claris/FileMaker since 1986

Did the initial work on many versions

- Microsoft Windows 3.0
- FileMaker Server



I started work at Apple in 1986 transferred to Claris when it was created in 87.
In 1989 I started first working on the FileMaker port to Windows.
Lead designer on the first version of Server, IWP, Draco, and Go.

Who is Clay Maeckel?

Been working at Apple/Claris/FileMaker since 1986

Did the initial work on many versions

- Microsoft Windows 3.0
- FileMaker Server
- Instant Web Publishing
- Draco Engine



© 2016 FileMaker, Inc.

#FileMakerDevCon

I started work at Apple in 1986 transferred to Claris when it was created in 87.
In 1989 I started first working on the FileMaker port to Windows.
Lead designer on the first version of Server, IWP, Draco, and Go.

Who is Clay Maeckel?

Been working at Apple/Claris/FileMaker since 1986

Did the initial work on many versions

- Microsoft Windows 3.0
- FileMaker Server
- Instant Web Publishing
- Draco Engine
- FileMaker Go
- Tractor Engine Repair



I started work at Apple in 1986 transferred to Claris when it was created in 87.
In 1989 I started first working on the FileMaker port to Windows.
Lead designer on the first version of Server, IWP, Draco, and Go.

Contents of this Session

Brief Draco engine history

How data is stored and downloaded

The file opening process

Draco changes made for FileMaker 15

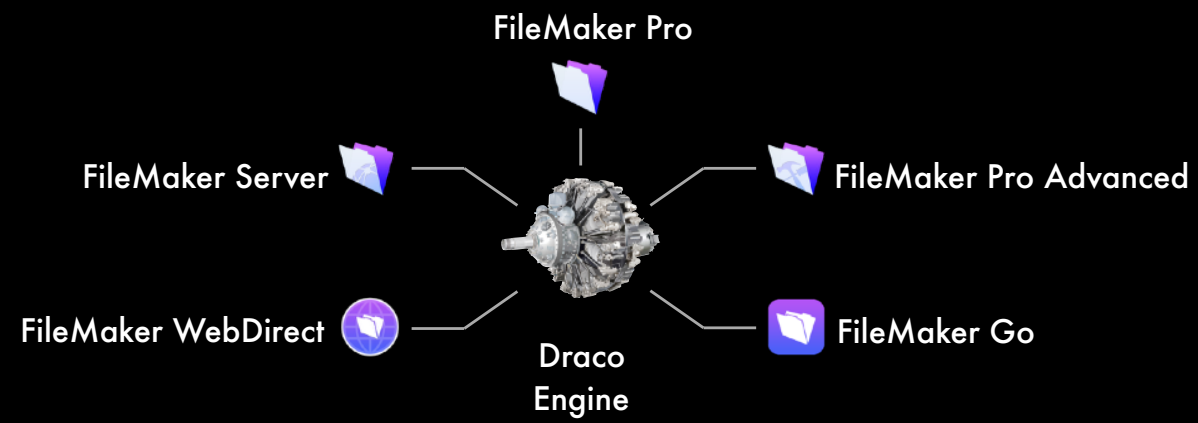
Q & A

Brief Draco engine history

© 2016 FileMaker, Inc.



The FileMaker Family



© 2016 FileMaker, Inc.

#FileMakerDevCon

Seen this many times, why an internal combustion radial engine, I don't know

Five Eras of the FileMaker Engine

1985 - DOS and pre-Claris

1988 - Flat file

1995 - Early relational

2004 - Draco based

2015 - Design surface

© 2016 FileMaker, Inc.

#FileMakerDevCon

Last year unconference and at Pause on Error I've showed some old versions of FileMaker

DOS and Pre-Claris

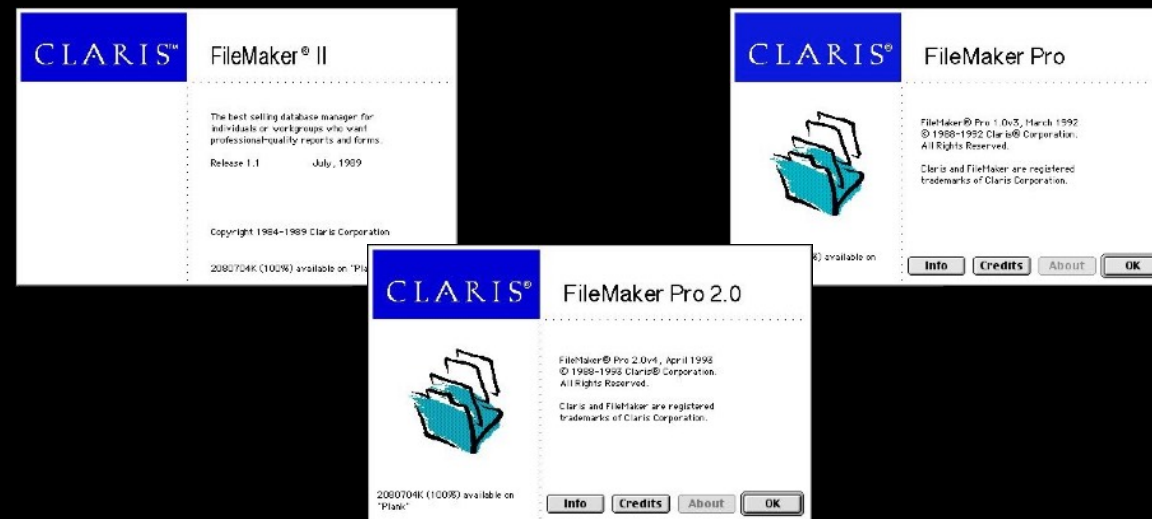


© 2016 FileMaker, Inc.

#FileMakerDevCon

'85 – flat file with lookups, one table per file, no or limited scripting
HBAM (talk about later) and peer-to-peer networking at HBAM api

Flat File



© 2016 FileMaker, Inc.

#FileMakerDevCon

'88 – ScriptMaker, MacDraw style layouts, Server, Windows version, Runtime, Spectre engine started with Draco layer to bridge the difference

Early Relational



© 2016 FileMaker, Inc.

#FileMakerDevCon

'95 – Pascal to C++, ODBC, IWP, plug-in architecture, Spectre cancelled and Draco expanded to entire engine, Steve Jobs came back and changed from Claris to FileMaker, Inc.

The Draco Team

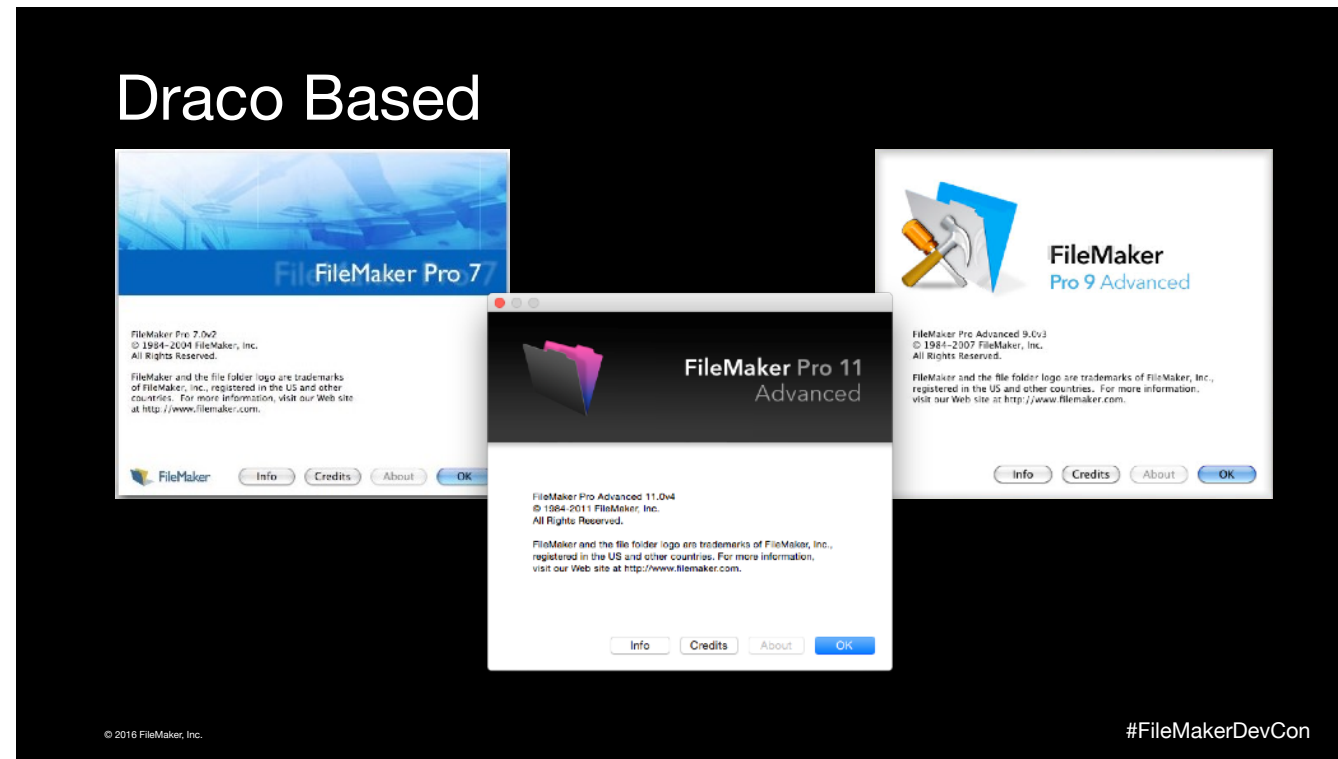


© 2016 FileMaker, Inc.

#FileMakerDevCon

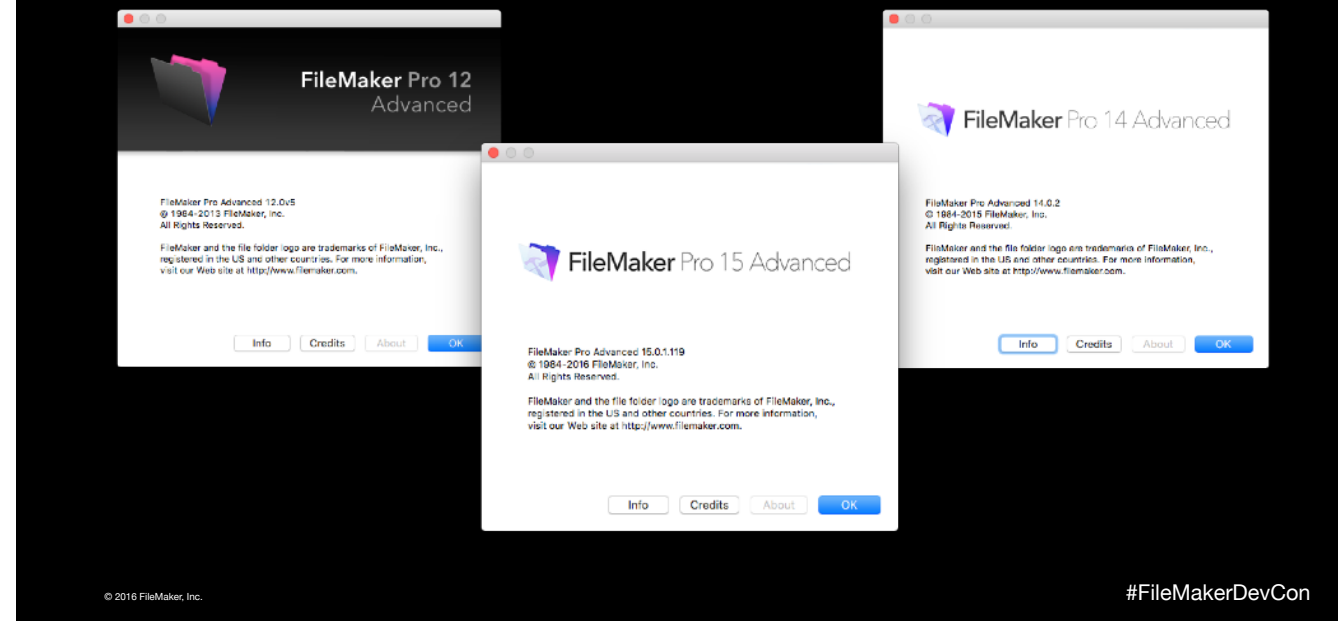
Jon is speaking later this week, Keith was the speaker at last weeks SF Bay Area FileMaker User group meeting, and me still here. Now have shown this slide on four continents, time to retire it.

Draco Based



'04 – the start of the Draco based products. Engine was designed for the hardware of the late 90's so when the iOS version came out in 2010 the engine ran fine

Design Surface



'15 – Move away from MacDraw style to CSS architecture. Unification of the Pro and Go drawing logic, move from IWP to WebDirect, iOS SDK

Draco Components

Support (OS abstraction)

© 2016 FileMaker, Inc.

#FileMakerDevCon

Mac (68K/PPC/Intel32/Intel64), Win (Intel32/Intel64), iOS (Arm6,7,7s,64), and others

Draco Components

HBAM (logical to physical representation)

Support (OS abstraction)

Design from '88, 512/1024/4096 block versions

Draco Components

DBEngine (calc, network, query, transaction)

HBAM (logical to physical representation)

Support (OS abstraction)

Typical database operations, where the SQL to FileMaker translation occurs

Draco Components

Design Surface (in progress, layout & draw)

DBEngine (calc, network, query, transaction)

HBAM (logical to physical representation)

Support (OS abstraction)

Now includes most of the PDF generation code as of 15 since we have moved the the DLI to Hummus libraries.

Draco Components

FMEngine (layouts, scripts, user model)

Design Surface (in progress, layout & draw)

DBEngine (calc, network, query, transaction)

HBAM (logical to physical representation)

Support (OS abstraction)

FileMaker Components

Draco Engine:

Protocols:

— HTTP
— THRIFT
— GLOP
— OTHER

Technology:

Native
Java
Generated HTML
PHP
3rd Party

FileMaker Clients

Native Code Client
ODBC Driver

Java Client
JDBC Driver

UAC Client

PHP Client

XML Client

Pro/Go Client

WebDirect Client

XDBC Listener

Web Server (Apache or IIS)

FileMaker Server

Tomcat
Vaadin
Admin Mgr

PHP
PHP2XML

Tomcat
XML
Vaadin
JWPC

Standby

Backup

SASE

CWPC

Database Server Processes (helper and server)

server upload, container optimization
main FileMaker communications channel

© 2016 FileMaker, Inc.

#FileMakerDevCon

Busy chart of all the components

Draco Usage

Draco Engine:

Protocols:

GIOP

Technology:

Native

FileMaker Clients

Pro/Go Client

XDBC Listener

FileMaker Server

Standby

SASE

CWPC

Database Server Processes (helper and server)

© 2016 FileMaker, Inc.

#FileMakerDevCon

Where Draco is used

How data is stored and downloaded

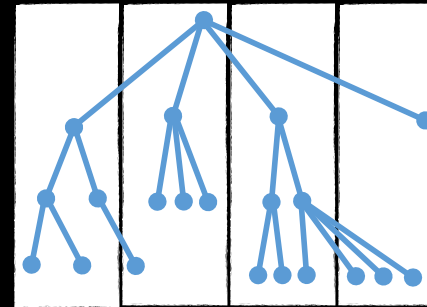
© 2016 FileMaker, Inc.



Hierarchical Tree File Format

The “branches” and “leaves” are divided into four kilobyte content blocks. These blocks are what the cache stores in memory

Most Draco operations consist of creating, moving, and deleting these branches and leaves



© 2016 FileMaker, Inc.

#FileMakerDevCon

The FileMaker “file format” is actually just a definition of what branches and leaves that have to exist in a large hierarchy. There is one branch for all layouts with a branch for each individual layout under it. The leaves of a layout consist of things like layout name, layout theme, a list of layout objects, etc.

The hierarchy is split up into a series of 4K blocks. These blocks do not have any fixed order and operations like save a compacted copy or recover can move these blocks around quite a bit.

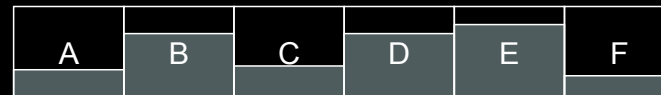
When you are adjusting the size of the “File cache” in the Pro product or the “Database RAM cache” in Server, you are adjusting how many of these blocks from all the files that are open that can be stored in memory. Blocks that are stored in the cache have been decrypted and optimized for faster inter-block access.

Most operations in Draco consist of manipulated the branches and leaves. The contents of many leaves of one branch can also be streamed to and from the FileMaker file thus not requiring the entire contents of the data to be loaded into memory at once.

FileMaker Blocks

In addition to the logical to physical mapping, HBAM manages the block order, free space, and encryption

Blocks before compacting (gray indicates used space in blocks):



Blocks after compacting (Save As Compacted command):

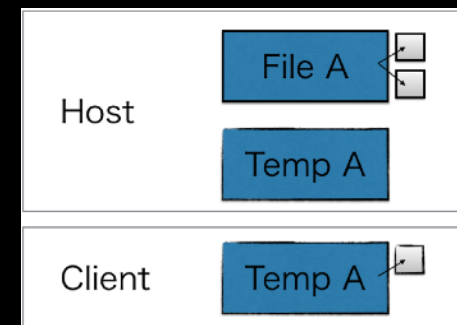


Also the “file cache” is storing in memory versions of these blocks, slide from ‘07

Temporary Files

Both the server and each client has a temporary file that contains a portion of the main file

External container data files or thumbnail representations of them are downloaded to a cache folder on the client



Both the host and client have temporary files associated with each FileMaker file it has open. Usually the temporary files contains what is currently being displayed while the main can be in the process of being updated by other clients at the same time.

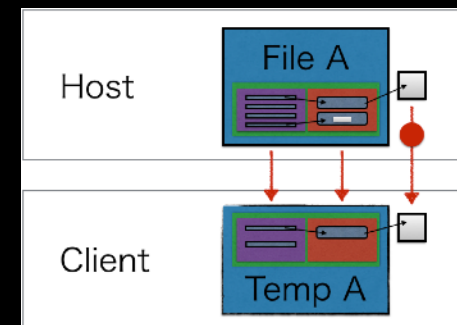
Browsing Operations

Entire records are downloaded

Containers download independently if needed

Thumbnails created on host

Globals and variables can be sent to the host



So when a network client wants to display a specific layout, it will request from the server all the contents of a specific branch of the “tree” and store it in the client’s temporary file.

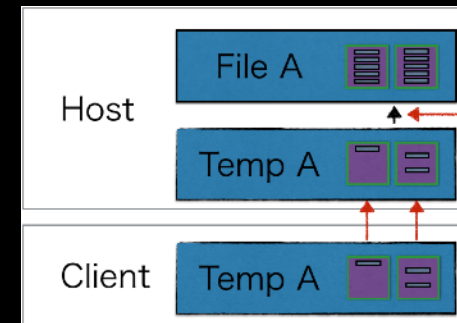
When a branch in the main file is finished being modified by any client, other clients are notified that that branch was modified and that the client should delete it from their temporary file as soon as possible. If the data changed is represented on the screen of a client it may then immediately download and update the display for what the branch represents.

External container data files work slightly differently but clients do keep a cache of most recently used one in a folder on the clients computer.

Committing Changes

Changed data is uploaded to the host's temp file first

Final network call to tell the host to commit changes to main file and index the changes



Applies to data and “schema”

More About Temporary Files

Always encrypted, level of encryption based on the main file's encryption

If created for a networked file, kept for 15 days otherwise deleted when closed

If free space on the disk is less than 250MB then the items in the temp files will be deleted if not currently in use

If free space on the disk is less than 2GB (1GB on iOS) then external container data is only cached while used

Temporary files to be deleted are in the system's temp folder, ones that are kept around are in the system's cache folder, change in version 15

The file opening process

© 2016 FileMaker, Inc.



Seven Steps to Open a File

1. Path list processing
2. Find the file
3. Create or reuse the temp file
4. Login
5. DBEngine processing
6. FMEngine processing
7. After opening

Note that not all opening processes and product go through all these different operations

Step 1: Path list processing

Process the list of paths to create a list to try

- a. Replace local/global variables
- b. Turn relative paths into full paths
- c. Remove paths that are not valid for platform

Step 1: Path list processing example

Before processing (script in file:/Mac/Work/invoice.fmp12)

```
file:lineitems.fmp12
```

```
$it
```

```
filewin:/C:/mystuff/lineitems.fmp12
```

```
filemac:/Mac/MyStuff/lineitems.fmp12
```

May have add a fileunix: for posix style paths, remember this is an old product

Step 1: Path list processing example

Before processing (script in file:/Mac/Work/invoice.fmp12)

```
file:lineitems.fmp12  
$it  
filewin:/C:/mystuff/lineitems.fmp12  
filemac:/Mac/MyStuff/lineitems.fmp12
```

After processing (on a Mac, \$it = fmnet:/clay.com/lis.fmp12)

```
file:/Mac/Work/lineitems.fmp12  
fmnet:/clay.com/lis.fmp12  
file:/Mac/MyStuff/lineitems.fmp12
```

May have add a fileunix: for posix style paths, remember this is an old product

Step 2: Find the file

Walk through the processed list of paths

- a. Check if the file is already open. Use the file if it is already open
- b. If fmnet, check for existing connection to host. If none, create a connection to the host
- c. Try to open the file for read/write access. If can't open, try again with read only access
- d. Continue to next item in list if nothing found yet

Step 3: Create temp file

- a. For local files create a random temp file name, for hosted files create a unique name for that file, host, and client
- b. If the temp file exists then reuse it **(NEW TO 15)**, otherwise create a new empty temp file
- c. Determine if any data currently in the temp file is out of date and delete any data that is out of date
- d. Copy versioning data, window location, auto login info, and locale settings down into the temp file

Richard and others excited about this

Step 4: Login

- a. If reconnecting, try the credentials that succeeded last time
- b. Try account/password from external source
- c. Try the parent file's credentials
- d. Try the file's auto login credentials
- e. On Windows, try the user's single sign on credentials
- f. Try the keychain/credential manager stored credentials
- g. Handle expired password case
- h. Ask the user to enter an account name and password

External source: ODBC or CWP login

Step 5: DBEngine processing

- a. If remote file download the following into the temp file
 - List of tables (but not their field definitions)
 - Table occurrences and their relationships
- b. Merge the new table occurrences and their relationships to other files and tables into the master map of all table occurrences.
- c. Some opening sequences stop here if they don't require a window to be opened. ODBC is an example.

Step 6: FMEngine processing

- a. Create window and the active object
 - For server side scripting it is a virtual window
- b. If remote file download the following into the temp file
 - The layout and its style sheet
 - Value list definitions
 - Font id mapping
 - Custom menus
- c. Queue any script triggers based on the opening of a window to run at the next available moment

This is an area we want to work on to remove items that have to be down loaded but it is a bit tricking to do without changing the file format

Step 7: After opening

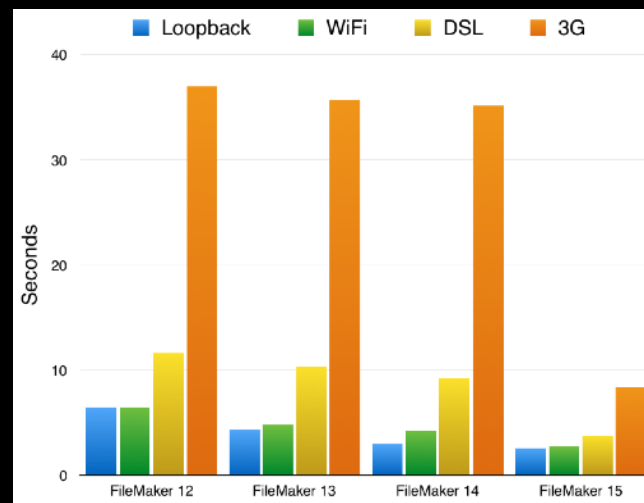
- a. The operating system will ask to draw the window
- b. Next available idle event will allow the scripts to start
- c. In the remote case more items are downloaded from the host when first referenced, including opening more files
 - Field definitions and the list of records
 - Records
 - Scripts
 - Container objects

Draco changes made for FileMaker 15

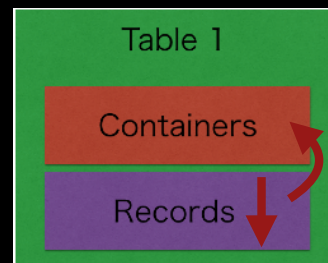
© 2016 FileMaker, Inc.



WAN File Opening Improvements



Delete All Records



Delete All Records – Truncate Table



Under Records menu when all records are selected, otherwise it says Delete Found Records

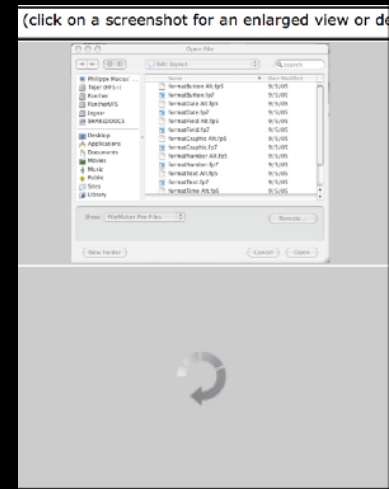
When Delete All Records Uses Truncate

- a. Table is local, or if hosted, we are talking to a server that can do the truncate table command
- b. The table is a FileMaker table and not from ODBC
- c. Trying to delete all the records, not just a subset
- d. The user has the permissions to delete all the records and there is no row level authorization calcs for delete
- e. There are no cascading delete operations for this table in any file
- f. There are no global container fields
- g. We can get a lock on all the records at once

Client Side Multi-Threading

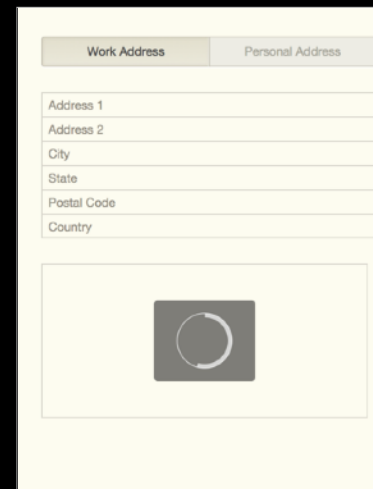
First was container field loading

You don't have to wait for large
image downloads or thumb-nailing
code to complete



Client Side Multi-Threading

Version 15 adds the portal in-line progress bar

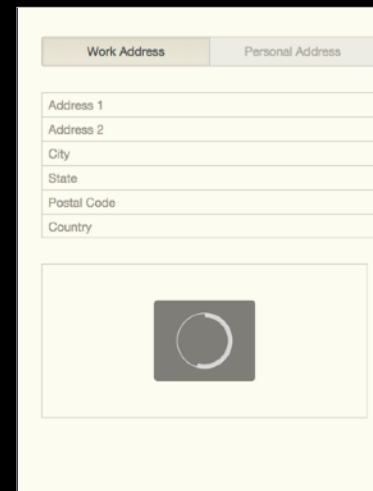


The screenshot displays a FileMaker portal interface. At the top, there are two tabs: "Work Address" (selected) and "Personal Address". Below the tabs, there are six text input fields labeled "Address 1", "Address 2", "City", "State", "Postal Code", and "Country". At the bottom of the portal, there is a large rectangular area containing a circular progress bar, which is currently at approximately 75% completion.

Client Side Multi-Threading

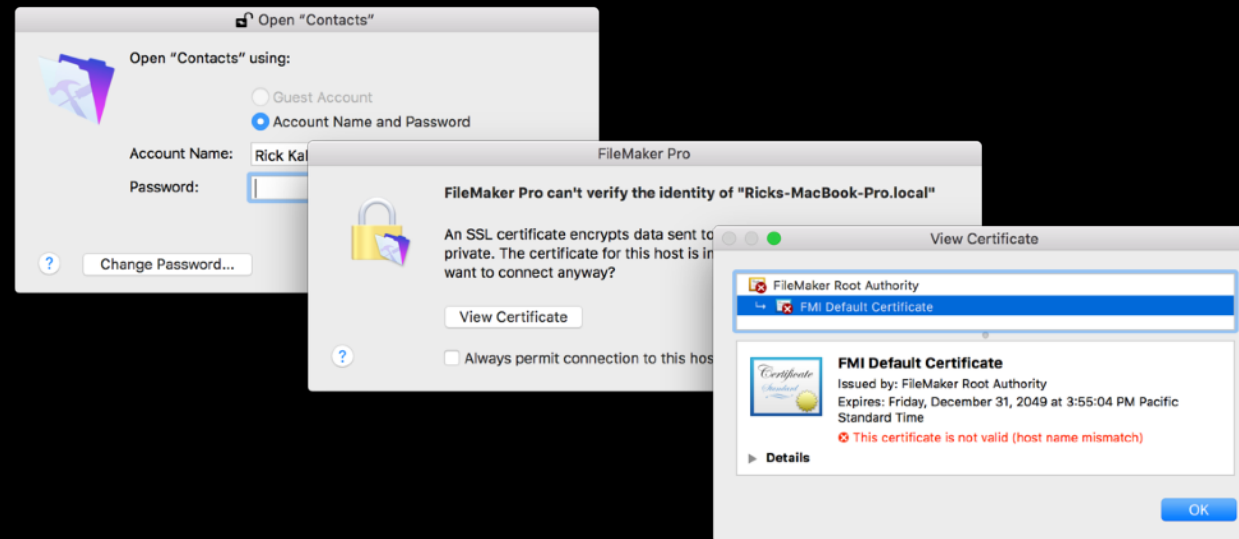
Version 15 adds the portal in-line progress bar

Don't assume that variables in Let() functions may be set in any particular order while a layout is drawing

A screenshot of a FileMaker layout. At the top, there are two tabs: "Work Address" (selected) and "Personal Address". Below the tabs is a portal containing a list of address fields: "Address 1", "Address 2", "City", "State", "Postal Code", and "Country". Below the portal is a large rectangular area containing a circular progress bar, indicating that the portal is loading data.

Talk about variables in layout objects

Proactive Security Warnings



Server Changes

See Jon Thatcher's "Under the Hood: Server Performance" on Thursday at 9:00 AM in Belmont 4/8.



© 2016 FileMaker, Inc.

#FileMakerDevCon

He will discuss more about FileMaker's networking and server processing, including how to use the new Server 15 feature "Top Calls Logging"

Session Updates

This session WILL have updates.

www.filemaker.com/devcon/speaker_updates

(This is also listed in your conference guide)

Please edit this slide to reflect whether you will or will not have updates to your session.

Q & A

Please use microphone so everyone can hear your question!

Thank You!

Please remember to fill out the session evaluation at:
www.filemaker.com/devcon/evaluations

You can also use DevCon2Go:

Under the Hood: Draco Database Engine			
Core - COR002			
Clay Maeckel (FileMaker, Inc.)			
Belmont 4/8 - Wednesday 3:45 PM			
Description	Objective	Background	More...



© 2016 FileMaker, Inc.

#FileMakerDevCon

1) If you have already completed more than one survey, you may have noticed that it was populated with previous information. If so, this was an indication that you were overwriting your previous response. The issue was fixed at 5:30pm on Tuesday, so we have your most recent response as of that time -- but could you please re-enter any surveys for you completed prior to it? The DevCon team apologizes for the inconvenience.

2) We're publishing a list of DevCon blogs in the FileMaker Community. If you'd like to see this list, or make a comment calling attention to yours, please go to the DevCon 2016 area of the community:

https://community.filemaker.com/community/devcon_2016