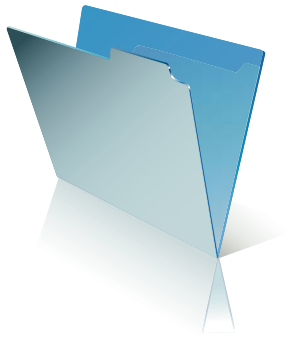
Technology Brief



Upgrading to FileMaker 7:

Migrating Existing Solutions

About This Technical Brief

It is the intent of this technical brief to help the experienced FileMaker developer better understand and prepare for migration to the FileMaker 7 product family. Reading this document will assist you in planning, preparing for, and executing a strategic approach to migration. Authored by Danny Mack, FileMaker Solutions Alliance Partner and President of New Millennium Communications, Inc., this paper is part of a series of technical briefs written by developers for developers, to assist in migrating existing solutions to the new FileMaker 7 product family.

For additional technical materials, please refer to printed and electronic manuals and online help that ship with FileMaker Pro 7, FileMaker Server 7, and FileMaker Server 7 Advanced.

Foreword

FileMaker® Pro is used in many ways, some quite simple and others very complex. This document is most applicable to developers who have built custom or commercial FileMaker solutions for their customers, database administrators who manage custom in-house solutions, or for "subject matter experts" who have become professional developers and who preside over collections of FileMaker files which are important parts of the business systems of an organization.

The FileMaker 7 family of products is the largest upgrade to the product line in the history of the company. Remarkably, it extends the ease of use and power of the application in both directions – making it easier for new users and less experienced database developers to create powerful solutions, while enabling more sophisticated developers to create enterprise-scale applications, handling large amounts of data and complex business processes, with major enhancements to the application model, the relational model, security, network performance, and web publishing.

FileMaker Pro 7 makes it possible to develop solutions much more rapidly and reliably, and to easily modify and extend the power of existing solutions. It is the ideal tool for building databases in a rapidly changing environment – for organizations that are innovating and evolving based on a strategic vision, and/or that need to respond to change quickly, flexibly, and economically.

Table of Contents

Introduction

This document is an orientation to migrating existing FileMaker Pro solutions to FileMaker Pro 7. The radical new capabilities of FileMaker Pro 7 necessitate some changes in the behavior of the application –making the migration process more extensive than the transition between any prior versions of FileMaker Pro. However, the rewards are proportionally greater as well.

Migrating your solutions to FileMaker Pro 7 needs to be considered from multiple perspectives. Not only will the new power and flexibility of FileMaker Pro 7 impact the way you develop, it will also have a significant effect on the problems an organization chooses to solve and how it solves them. Including different organizational perspectives in the migration process not only benefits the migration; it helps prepare the entire organization to take advantage of the new capabilities of the FileMaker 7 platform in the future.

This document helps a developer and decision-maker to have an informed basis for making decisions about migration. For some solutions it is possible to simply convert the files to the new FileMaker Pro 7 format and run. **However, in many cases, it is necessary to modify your files to be able to replicate the original behavior of the solution. It is optimal for some of these modifications to be performed in FileMaker Pro 6 *before conversion* as well as in FileMaker Pro 7 after conversion.** In other instances, developers will want to entirely rewrite solutions to take advantage of the new capabilities and to simplify ongoing development.

This document does not cover these issues completely; especially not all of the technical issues, and other important reference materials are listed in Appendix 1 of this document. The document is based on three overarching premises:

1: Learn FileMaker Pro 7 before you convert an existing solution.

FileMaker Pro 7 is more different from prior versions of FileMaker Pro than you expect! Create some simple databases and explore the new features. Look at the example files that come with FileMaker Pro. Do trial conversions of your solution and explore and test in depth before considering final conversion and "going live".

2: Know your solution and your business.

There are many technical issues to consider before migrating to FileMaker Pro 7. Not all of the issues affect all solutions, so it is important to be familiar with your solution and able to assess which issues are relevant. Analysis tools are available to assist in doing this. (See Appendix 2 for more information.) Furthermore, different approaches to migration present different opportunities and challenges. Making the right decision for your situation has as much to do with your business and its goals, as it has to do with FileMaker Pro and specific technical issues.

3: Create multiple stable points in the migration process.

It is unlikely that you have an unlimited budget, and that you have end-users or business owners who care more about reliable access to the data in your solution than they do about the optimal design of the system. Rewriting a solution can take a long time. It may be practical to convert a solution and get it working as intended, and then later add new functionality in phases.

Accepting these premises, and applying the strategies in this document, will greatly enhance your understanding and experience of the migration process. The condition and complexity of a solution as well as the expectations for its future use and evolution determine which migration approach is most likely best. An optimally designed FileMaker Pro 7 solution is quite different from a FileMaker Pro 6 solution. If you don't develop a clear idea of where you're going and how best to get there before you begin your migration, you won't end up where you intend to be… at least not very soon.

This document is an overview of the issues and approaches to migration. The *Migration Foundations and Methodologies* document includes a methodical approach to migration, including a detailed review of the technical issues. Additional relevant information is available at http://www.filemaker.com/upgrade/techbriefs.html, and in the associated Tech Briefs on FileMaker Server, Security, and Web Publishing.

Terminology

To get the most out of this document, the reader should be familiar with the following:

Migration	The process of evolving or rewriting a FileMaker Pro 6 (or earlier) database solution through the stage when it is successfully deployed in FileMaker Pro 7.
Conversion	The technical event of converting files from .fp3 or .fp5 format to .fp7 format, as performed by FileMaker Pro 7 or FileMaker Developer 7.
Table (Data Table)	A collection of data organized in columns and rows. The rows are records and the columns are data elements. FileMaker Pro 6 allows one table per file. FileMaker Pro 7 allows multiple tables per file.
Table Occurrence	An instance of a Data Table used in the Relationships Graph. Each Table Occurrence has its own unique name and relationships.
Relationship	A relationship in FileMaker Pro identifies the data fields and operators in two Data Tables used to relate or "join" the tables. FileMaker Pro relationships also define rules of dependency and creation.
Schema	The collection of tables, relationships, and fields which make up the data structure of a database.
Graph (Relationships Graph)	The graphical representation and workspace of Relationships and Table Occurrences in FileMaker Pro 7.

Logic (Business Logic)	Rules coded into a solution that define how data is entered and manipulated. In FileMaker Pro, the logic is coded into scripts, calculation fields, value lists, relationships, and layouts.
Process (Business Process)	A feature of a solution that facilitates a particular action. The process may be embodied in scripts and layouts, but is usually referred to from the point of view of users of the system. It is not limited to "business" in the sense of "commerce" – it could be referring to creating an invoice, printing a student transcript, or entering results of a laboratory experiment.

Migration – An Overview

Convert or Rewrite – Is that the question?

There are compelling reasons to convert a solution: there may be a large existing investment in programming hours, testing, embedded business logic, and familiarity with its use by many people. Some of the business logic may not be known, or at least not remembered, by those who are now managing the database. There may not be sufficient time nor budget to rewriting the solution.

There are compelling reasons to rewrite: FileMaker Pro 7 has many advantages that can most efficiently be realized by building a solution in FileMaker Pro 7 from the beginning. The old solution may be less than ideal in the first place, containing obsolete or poorly implemented features. There may be a fair amount of effort and diligence required to convert an existing solution, and it may be more productive in the long run, and, frankly, more enjoyable, to build from scratch.

A hybrid approach is possible, in which all or part of a solution is converted, and then part of the solution is rewritten or new files are developed and then integrated with the converted files. It is important to clarify the expectations associated with upgrading before choosing an approach to migration. Understanding both the consequences of conversion, and the various options for migration, is essential to determine which methodology is best suited for your solution.

Why Upgrade? – Immediate User Benefits

There are several benefits to converting to FileMaker 7 that provide immediate benefits without any modification to the files (unless, as in some cases, modification is required to restore previous functionality).

No practical file size limit – The file size limit is 8 terabytes, but it is not possible to reach that on current operating systems or available storage media. The limit of FileMaker Pro 6 databases was 2 GB, and if this limit was reached a file could be damaged. This new capacity allows FileMaker Pro 7 to be used for very large databases with many records, and with container fields containing large images or other documents.

No file count limit – The limit of 50 files per user at one time was a serious constraint for customers with large, complex solutions. Developers can now concentrate on optimal solution design without the artificial constraint of a limited file count. (However, FileMaker Server is still limited to 125 files being hosted per server.)

There will be somewhat less of a need for a large number of files now that each file can contain multiple tables, with a limit of one million tables per file.

Container Fields – In FileMaker Pro 6 and earlier, container fields could contain only a few types of media files. There are now more types that container fields can display Most dramatically, though, container fields can now contain ANY file type as a binary object, up to 2 GB per document (the limit of an entire file in FileMaker Pro 6!). A document stored as a binary displays as an icon, and to access the document it is necessary to export it from the container field. However, a document can now be stored as a file reference only, and the file references (file paths) can be manipulated as text without having to "re-insert" a document, which makes FileMaker an extremely powerful tool for document management.

Huge and versatile text fields – The limit in the prior version was 64K. If this limit was exceeded in a scripted operation data could be lost. Text fields (as well as Container fields) now hold up to 2 GB. This allows FileMaker Pro text fields to handle large written documents, so that contracts, white papers, articles, or even books, can now be contained in a text field. The new text engine and enhanced text formatting features also help. Calculated text results now retain the text formatting, styles and colors of their sources.

More Server Power – FileMaker Server is radically enhanced and the client/server model significantly changed. More processing is done on the Server and it can now take advantage of multiple processors and additional RAM to scale performance to demand. Remote administration in a web browser via the SAT (Server Administration Tool) is elegant. Encryption of network traffic requires simply checking a checkbox and a restart of FileMaker Server 7. Network traffic is optimized so that LAN and WAN performance is considerably greater. The web publishing capabilities in FileMaker Server 7 Advanced is much more robust than in prior versions. See the associated Tech Briefs on Server and Web Publishing for more information.

Security – In addition to the Server-based security enhancements, the file format itself is much more secure. The database structure, including the data itself, is now encoded so that it is not readable by opening a file with a text editor. Furthermore, the privileges assigned in the access privileges interface are now extended consistently to all external API's. There are many other enhancements to security. See the associated Tech Brief on Security for more information.

Unicode – FileMaker now stores text in Unicode format, which means that it can store double-byte languages in the same fields with single-byte languages. For instance, a database can now contain both Japanese and English. This enables international sharing of applications and more rapid development of multi-language solutions.

Very Robust File Format – The method by which FileMaker Pro 7 and FileMaker Server 7 write changes to a database make it more rare that a file becomes damaged or that any corruption is introduced. Furthermore, the conversion process does not propagate corruption that may have existed in a prior version. This adds considerably to reliability and stability.

Precise Math – FileMaker now resolves calculations to 400 significant digits, and stores numbers with up to 400 digits to both the left and right of the decimal. This makes FileMaker Pro a better choice for scientific and engineering applications. Also, new methods for performing simple math which don't rely on built-in processor/OS floating point logic, minimizing the possibility of rounding errors.

Sticky Portals – No more "portal pop"! In earlier versions of FileMaker Pro, when a scrolled portal is exited by any means, including simply when going to another field on a layout, the portal "pops" or "snaps back" to the top row. Now all portals on a layout and can be left in a scrolled position at the same time. If a layout is formatted not to show active field frames, the portal sticks until FileMaker re-draws the layout, usually when the record is changed or when the user leaves the layout.

Much More!

Why Upgrade? – Developer Benefits

FileMaker Pro 7 offers remarkable new possibilities for developers. Developers are able to enter new markets, and start entirely new development business thanks to the new power of FileMaker Pro 7. Some features also provide benefits to end users, but usually require developer intervention, so they properly belong in this section.

Access Privileges – The access privilege model is new. FileMaker Pro 7 now supports Active Directory and Open Directory authentication. There is very sophisticated, comprehensive, and flexible management of user privileges, including much more granularity of access. There is more ability to manage security in a central interface, both because of the ability to put an entire solution into a single file, and because of new capabilities for scripting account management across multiple files. The granularity of access privileges, combined with the modularity of the "file" entity, allows new business models in which developers are able to protect intellectual property (database structure and logic) while allowing users and other developers to extend the features of their solutions. See the associated Tech Brief on Security and the *Migration Foundations and Methodologies* document for more info.

Web Publishing – The new web publishing capabilities of FileMaker Pro 7 and FileMaker Server 7 Advanced are addressed in the associated Tech Briefs on FileMaker Server 7 and FileMaker 7 Web Publishing. Notably, both Instant Web Publishing and Custom Web Publishing are dramatically enhanced. Developers can more easily extend the features of solutions to users via web browsers, and exchange data with other data sources via XML, ODBC, and JDBC. It is important to note that CDML is no longer supported, though FileMaker, Inc. provides a CDML to XSLT conversion tool with FileMaker Server 7 Advanced.

New Relational Data Model – The robust new relational model allows relationships to be much more powerful and flexible, accessing data multiple tables away without needing to "pipe" data through calc fields. This data can be accessed in calculations, and can be displayed and edited in portals. Relationships can use relative operators (greater than, less than, etc.) and multiple predicates (more than one match field per relationship). Also, when valid, indexable keys exist on both sides, relationships can be bi-directional. Globals in a different table are now accessed without a relationship being required. Most radically, the visual interface provided in the graph lets developers more easily see the logic and the structure of solutions, and the drag-and-drop interface allows MUCH faster design of the relationships. Communication among multiple developers and between developers and customers (or "business process owners" in the case of an in-house solution) is enhanced by the ability to collaborate with a visual interface, enabling more accurate and effective business process modeling and implementation.

Schema edits on shared files – It is possible to edit schema, including defining tables, fields, relationships, and even access privileges, while other users are connected to a hosted database. There are circumstances when this is very helpful, though in most cases it is advisable to maintain a separate development version from a deployed production system.

Multi-user testing on a single computer – In FileMaker Pro 7 it is possible to set up a session which is a guest of itself, so that it is possible to design and test in a multi-user environment with only one computer.

Flexibility and Efficiency – One of the many advantages of FileMaker Pro 7 is that it now imports scripts and pastes layout objects more reliably, resolving all references by name. This means that it is possible to import scripts from file to file, or to copy layout objects or entire layouts from file to file or between tables within a file, and all references will be resolved correctly if the referenced scripts, fields, and table occurrences exist and are named identically. Furthermore, since two different layouts, based on different table occurrences, can have valid relationships to a common table occurrence, a portal or related field can be copied and pasted from one layout to another and it will copy correctly, but its context will change automatically, producing different (appropriate) data results. When copying and pasting layouts, everything comes over except for tab order. To retain tab order, layouts can be duplicated and then their context (table occurrence) can be changed if desired… one of the many benefits of having multiple table occurrences in a file.

Editable File References – In prior versions of FileMaker Pro, file references are not displayed. They are stored in files, but without an interface to see or to edit them directly. It is possible to view the file references in a FileMaker Pro 6 file, and to prepare a FileMaker .fp3 or .fp5 file for conversion to FileMaker Pro 7, by using third-party tools such as MetadataMagic from New Millennium Communications. File references control where and how files look for other files. File references can contain a full path, a relative path, and/or a network address. File references are used by relationships in FileMaker Pro 6, certain script steps (including Perform Script [External], Open, Close, and others), and by external value lists. The rules, which earlier versions of FileMaker Pro used to evaluate file references and to locate other files, are somewhat obscure, and the rules varied in different versions of FileMaker Pro. File references in FileMaker Pro 7 are now editable as a text block in a dialog, and can contain a list of references, with a simple syntax so it is easy to point one file at another and to create your own rules, a "hunt order", which FileMaker follows when it looks for files.

New Calculation functions – There are MANY new calculation functions in FileMaker Pro 7, among them powerful new text manipulation functions, including calculations which operate efficiently on lists in a text field, and others which modify text styles and colors. Among the most powerful new functions are "Evaluate" and "Let". The Evaluate function puts calculation logic into a text field, and FileMaker Pro evaluates it as if it were a native calculation. Evaluate(FieldName), in which the field contains the text "2+2", will return 4. This allows developers to provide an interface in which users can define custom logic. The Let function allows the defining and naming of a variable within a calculation, so that the variable can thereafter be referenced in the calculation by name, useful when a calculated value is used multiple times in a larger calculation, with benefits for both convenience and readability of calculations.

Custom Functions – In FileMaker Developer 7, it is possible to define one's own calculation functions and store them in a file. These calculation functions thereafter appear in the list of available functions in FileMaker Pro 7 or FileMaker Developer 7, like other functions. Notably, custom functions can be recursive, so it is possible to do things in a custom function that aren't otherwise possible in a FileMaker Pro 7 calculation. Custom functions need to be added to each file using FileMaker Developer 7. Custom functions allow developers to add their own features to FileMaker Pro 7.

Script Parameters – Scripts which are triggered by a button or called as a subscript can now be provided a parameter, associated with the specific button or specific Perform Script step, and that parameter can be

evaluated at any point in the script (and can be passed on to subsequent subscripts if desired). This allows very much more efficient scripting, and many fewer scripts. It is no longer necessary to have many, nearly identical, scripts to perform a similar process on multiple tables. It is now possible to have one script called from different buttons that performs an identical action but with a different parameter value (very useful in a calendar, for instance).

Commenting Calculations – It is now possible to comment calculations with C-style or C++-style comments. This makes code more understandable for other developers, as well as for the original developer when they come back to it at a later time. It also makes it possible to "comment out" parts of a calculation, an essential programming efficiency.

Developer interface for Find, Sort, Print Setup, Import, Export – In previous versions, these five script steps were able to store their settings, but those settings had to be defined first by a user action in FileMaker Pro, and then the script step settings could be saved, or later replaced (overwritten) if desired, by saving the script. In FileMaker Pro 7, these script steps now have an interface in ScriptMaker™, allowing for much more efficiency, readability, and ultimately reliability, of script logic and behavior.

Much More!

FileMaker 7 Migration Fundamentals

A file is a file, or is it? – the new metaphor

Designing solutions in FileMaker Pro 7 is very different from designing in earlier versions. This is largely due to a fundamental change in the concept or "metaphor" of the file.

In FileMaker Pro 6, and in all earlier versions of FileMaker Pro, a file contains one table of data only, and it can be presented in one window only. The same file contains the scripts and layouts with which you interact with that data: the "logic" and "interface". Since FileMaker Pro became relational, beginning in FileMaker Pro 3.0, it has been possible to display related data in another file. The fields accessible, however, are only one table away, and your perspective on the data is always based on the file that you are in.

One fundamental implication of this model is that local use of data in another file is always "unstored", and the functionality of local relationships and performance of local calculations is based on that, with substantial implications for solution design.

In FileMaker Pro 7, a file may contain many tables, or no tables. A file may contain interface elements (layouts and scripts), or no interface. A file may or may not contain logic (calculations and scripts). A file may be presented in many windows that are open simultaneously – and the windows can be created and dismissed by either end-user actions or by scripts.

Data in another file is no longer inherently "unstored" from the perspective of the file that you are in. A layout in a given file can now display data from a table in another file, just as if that data resided in the local file (in a list view, for instance). Local use of data from another table still has the same constraints as in previous versions

as it is unstored, but a layout that displays data from another file behaves identically to a layout in the same file that contains the data. Similarly, scripts which manipulate data in another file behave as if they were in the same file.

Design Options

What are the implications of these new capabilities? There are now a broader range of possible designs for your solutions. These options include:

1. A single file solution, with many tables in one file, and with all interface and logic (layouts, scripts, calculation fields) in the same single file as well. There are considerable advantages to this approach for certain types of solutions, especially for small solutions built by a single developer.

2. Separation of data and interface, with all data tables in one file, and with all layouts and scripts in a different file. This facilitates much easier upgrading of solutions and less frequent data imports – as well as providing a less fragile environment for deployed solutions, since it is more commonly practical to develop in an offline copy of the solution.

3. Multiple data files, and multiple interface files. There are many reasons why one would want to have multiple files – discussed in more depth below.

Migration Scenarios

The new capabilities of FileMaker Pro 7 have significant implications to the migration approach, and one methodology does not fit all. The optimal architecture and migration methodology depends on many factors and considerations, including: intended market and deployment (in-house, custom, or commercial), scale, the skill of the developer(s), and budget and time constraints.

Scenarios

1. Convert and run without modifications (this works in simple situations)

2. Convert and modify to make it work as before

3. Convert, modify to make it work as before, and enhance (add new features)

4. Redesign and rewrite (data imported later)

Methodologies

There are many approaches that can be taken. Some are more applicable to one situation than another. Some approaches produce a better result, and/or are much more efficient than others. These approaches are explored in depth in an associated document, *Migration Foundations and Methodologies*. Here's a quick overview:

1. Convert and Deploy, Then Modify if Necessary – this may work for simple solutions, ones that don't involve multiple files, relationships, scripted navigation, scripted data manipulation, field validations, or passwords. Even for moderately complex solutions with a limited implementation of the above elements, it may be possible to convert and then fix problems as they arise.

2. Modify, Convert, Test, & Modify – this approach recognizes that it is more efficient to make methodical changes in FileMaker Pro 6 to prepare files for conversion, and then to make fewer additional modifications after conversion. Iterative modification and testing, with repeated "dry run" conversions, is a more efficient and reliable process for migrating solutions than to convert first and later try to figure out what's not working.

3. Add a new interface file to a converted solution. It is possible to convert a multi-file solution and then add a new interface file to that solution, referencing the data tables in the converted files, thus enabling you to incrementally rewrite the interface.

4. Consolidate multiple files into a single file, starting with an existing file. This is most practical when a single file already contains a relatively large percentage of the total structure and interface of a solution.

5. Consolidate multiple files into a single file, starting from scratch. This is desirable when you intend to take full advantage of the new capabilities of FileMaker Pro 7, or where there is little benefit from starting with an existing file.

6. Create new files – an interface file and a data file, or multiple interface and data files – appropriately consolidating data tables and fields, as well as layouts and scripts, into the new files. Given the significant differences and the efficiencies of FileMaker Pro 7, it may or may not make sense to replicate existing calculations, relationships, and, especially, scripts, but be better to redesign a solution with the new possibilities in mind.

7. Co-existence of .fp5 and .fp7 files. There is no direct means for relating data in FileMaker Pro 5/6 files and FileMaker Pro 7 files. Therefore, some additional means of data exchange is required to make this scenario work. Sharing data between files in the two different formats may be a requirement for certain situations to enable incremental migration of a large multi-file solution, or because some files or data are "owned" by different departments in an organization, and not all departments are prepared to upgrade at the same time.

Conversion Basics

What FileMaker Pro 7 does when it converts your old files

FileMaker Pro 7 doesn't actually take the old files and change some things in them. Instead, it reads those files and then writes entirely new files, recreating and "translating" your field definitions, relationships, scripts, and layouts to the new format, and bringing over your data as well (and recreating the indexes) if you have data in the files when you convert them.

One benefit of this is that file corruption is not propagated to the new version. On very unusual occasions, a damaged field or script or layout element may be lost, but the new files are "virtually clean".

Many elements of your solution are changed in the conversion process. The changes, along with much other useful information, are documented in the .pdf file entitled "Converting FileMaker Databases from Previous Versions", which is provided on the FileMaker Pro 7 and FileMaker Developer 7 product CD, and a current version of which is available for download from http://www.filemaker.com/upgrade/migration.html.

An overview of changes:

File References which are invisible in FileMaker Pro 6 and earlier (except by using third-party tools such as New Millennium's MetadataMagic) are now displayed, and the format of file references is considerably different.

Relationships work very differently in FileMaker Pro 7 and they are represented entirely differently. Relationships are now bi-directional, and they can link via "table occurrences" to tables more than one "join" away. Notably, relationships in FileMaker Pro 7 are represented only graphically – there is no "text only" view in the FileMaker Pro 7 design interface.

When a multi-file solution is converted, each file has its own relationship graph, representing the relationships that exist in that file only.

Field Definitions – The names of many calculation functions changed, and in some instances the syntax changed as well. The conversion process translates old calculations to the new format. Some calculations changed behavior, though most (but not all) of the changes are innocuous. See the "Converting FileMaker Databases from Previous Versions" document for more.

Script Steps – Similarly, some script steps changed behavior. Some changes are not innocuous, and have significant consequences in converted solutions. The conversion process does a good job of making some minor changes to scripts to attempt to emulate the prior behavior, with reasonably good success in most cases. In some cases, there is a need for additional manual modification of scripts to fully achieve the former behavior.

Layouts – In general, layouts are rendered faithfully after conversion. Fonts are now anti-aliased, which makes graphics and larger text render better, but may cause smaller text to appear fuzzy. The rendering or position of some objects may not be identical, with consequences particularly for overlaid and transparent objects.

Access Privileges in FileMaker Pro 7 are radically more sophisticated and powerful than in previous versions. The conversion process does bring over your access privileges, including a one-for-one conversion of your passwords (which are converted to "accounts"), but there are usually some significant changes to the structure of groups, which loosely translate to the new "privilege set" feature of FileMaker Pro 7. See the technical brief on FileMaker 7 Security) for additional information, available at http://www.filemaker.com/upgrade/techbriefs.html.

What FileMaker Pro 7 does not do when it converts your old files

FileMaker Pro 7 does not convert a multi-file solution into a single file solution. It is a common misconception on first impressions of FileMaker Pro 7 that it would be beneficial (and practical) for FileMaker to do this.

It doesn't do it because FileMaker Pro 7 can't make the necessary decisions for you. To take advantage of having all of your tables in one file, you will have many fewer relationships, fewer scripts, fewer or different calculation fields, etc. – and there will be compelling reasons to still have multiple files.

Given the extraordinary changes to the relational model, it is actually quite remarkable that the automated conversion to the new format has been made at all.

The Conversion Conundrum

Of the many ways to architect a multi-table FileMaker Pro 7 solution, one of the only ways you would not choose to do it is the way that your files are organized after conversion. You can put all of your tables and interface in one file, or separate your data tables and interface into two or more files, but you will not want to put each data file and its interface into separate files. Yet that is exactly the state of converted files.

Thankfully, solutions can work well in this configuration. And, thankfully, there are practical strategies for migrating a solution to a more optimal FileMaker Pro 7 architecture. By taking an incremental approach to converting and extending a solution, a cost-effective migration strategy can be realized.

Database Design and Evolution

It may be cost-effective and practical to find multiple points in the migration process when your solution can be stable and functional. This provides benefits to your organization if you are an in-house developer, and allows you to more quickly take advantage of FileMaker Pro 7 if you are an independent custom or commercial developer.

When re-writing a solution from scratch, there is still a conversion component. At a minimum, the old solution will likely be converted to FileMaker Pro 7 prior to importing data into the new solution. (It is possible to export data from a .fp3 or .fp5 file to a text file and then to import it, but that does not handle the contents of container fields.)

Most FileMaker Pro solutions evolve over time and are enhanced based on wisdom gained along the way – both from the insights of the developer and based on feedback from users. Developers' technical knowledge increases over time and that is often reflected in the incremental enhancements to the features of a solution.

The evolution of FileMaker Pro 7 solutions is similar. In fact, the expertise and knowledgebase which exists about FileMaker Pro 7 is far less mature, and it will take a while for best practices and true expertise to develop, so an incremental approach may prove to be the most reliable way to minimize what will prove in the long run to be novice errors.

Compounding the advantages of taking this approach is that needs change over time, so that by the time you finish a grand design the priorities may have changed. True business value is often realized by an incremental strategy to developing new systems.

Convert and Deploy – only for very simple solutions

You may have a sufficiently simple solution that it will be practical to just convert and run. However, it is not advisable to just "hope" that your solution will fit this scenario and will not require any modification to work correctly after conversion. If you do just convert, you may discover that you won't be able to launch your solution because of password issues or other conversion-related changes and you will realize that you need to be more methodical, and follow the instructions for pre-conversion tasks.

If you can get into your solution after conversion without a problem, you may notice that the layouts look pretty good and most of the features work, but don't just "deploy and cross your fingers". First of all, **make sure you have a good backup of your pre-converted files.** Work with your users and test extensively to see if all features are working correctly.

Most of the time the features work or calculations produce the same result as they did previously, but sometimes they don't. This can be a very big problem if you don't anticipate it or know where the problems may occur. That is one reason why it is essential to understand the issues in advance of conversion and to know what to look for. It is advised, at a minimum, to review the Conversion Issues and Resolutions section of the *Migration Foundations and Methodologies* document.

It is very impractical to "downgrade" to your old solution in FileMaker Pro 6 once you have current data in your new solution. You would need to export all of your data to text files and import it into the old files (assuming you have a backup!), and you would need to find some way to get the images out of container fields and copied back into the old solution.

Multiple Stable Points Phase 1: Modify and Get Working Reliably

There is a large but finite set of issues that need manual attention to enable converted solutions to operate as they did in FileMaker Pro 6. Not all of these issues affect all solutions. Many issues are best addressed prior to conversion.

An important aspect of the process of manual modification is testing. Depending on the scale of a solution, it may or may not be feasible to test all the features of a solution. Scripts may work perfectly reliably if performed in a certain sequence, but not if performed in a different sequence. It will be most efficient and reliable if you, the developer, understands the issues that need to be addressed and then systematically and completely evaluate, and modify if necessary, all instances of the issue in your solution. Analysis tools help to do this.

Some tools are available now and more tools will emerge. (See Appendix 2 for more info.)

The primary tasks:

- **Conversion of access privileges**
  - Case sensitivity of passwords
- **Scripting and calculation changes**
  - Navigation scripts
- **On Open and On Close scripts**
  - Several specific calculation result changes

- **Problems with file references**
  - Performance issues
- **Managing data integrity**
  - Changes in when and how records are locked and committed
  - New Record, Set Field, Validations
  - Consequences of multiple windows
- **Plug-in compatibility and interaction with third party products**

These issues, and what to do about them, are discussed in depth in *Migration Foundations and Methodologies*.

There are two fundamentally different approaches that can be taken to the conversion and modification process. One approach is to freeze all development in the old version, convert it, spend time modifying and testing it until satisfied, import data, test, tweak if necessary, and go live.

The alternative, and the approach documented here, is to do multiple "dry-run" conversions with or without data, making only minor modifications post-conversion, and at each iteration testing the reliability of the solution… until the conversion process is satisfactory. This allows for ongoing development and eliminates the need for an import of data in the final migration.

It is not possible with a tool that analyzes a .fp7 database structure, such as the FileMaker Developer 7 Database Design Report, to assess whether a converted solution is working as intended. However, by examining a FileMaker Pro 6 database, and knowing the behavior of FileMaker Pro 7 when it converts a database, it is possible to assess the "definite" and the "possible" issues which will be encountered so that they can be dealt with systematically. The analysis tools available in FileMaker Pro .fp5 format are mature and reliable.

The process looks like this (with the pre-conversion stage being considerably longer than post-conversion):

- **Pre-conversion**
  - Pre-conversion modifications
  - Dry-run conversion
  - Testing
  - (Repeat)
- **Post-conversion**
  - Post-conversion modifications
  - Testing
  - (Repeat)
- **Deploy (over the course of a few days?, weeks?, months?)**

Please refer to *Migration Foundations and Methodologies* for an in-depth documentation of this process.

Multiple Stable Points Phase 2: Enhance to Leverage FileMaker Pro 7

In *Migration Foundations and Methodologies* document, various techniques for efficiently and reliably realizing the advantages of FileMaker Pro 7 in converted solutions are discussed. It also addresses some of the technical challenges with re-writing solutions from scratch.

Beware of this very significant consideration:

A developer may be tempted to begin to add features to converted files. The problem with this approach is that as soon as you add a layout to a file which is based on a different table than the one for which the file was originally designed, you risk the scripts in that file operating on a different data set than the one intended, since they were not written with the anticipation of multiple tables in a file. As a result, the data integrity and the usability of the solution may be compromised. It may require lots of "retrofitting", primarily modifications to scripts, to make it work reliably. Also, this is not likely to be a satisfactory long-term approach because the design of the solution is not optimized for FileMaker Pro 7.

Instead, a new empty "interface" file may be created which can access the data in the old files as if it were in the same file. This is done by creating table occurrences in the new file that reference the tables in the old files. Now you can begin to add new features to the system without affecting the logic and interface of the old files. The old files continue to work as they did before and the new Interface file provides a framework for developing new FileMaker Pro 7-optimized features.

Since FileMaker Pro 7 makes it easier to move logic and interface from one file to another, since references are resolved by name, it may be practical to move part or all of the old interface into the new interface file over time. Similarly the developer may choose to move all of the tables from the old files into the new interface file or into other new data files. As the interface, logic, and data get moved over into a FileMaker Pro 7-optimized structure some parts may get left behind.

Following this method, a solution can be moved (at the appropriate pace) down a path of multiple stable points from a FileMaker Pro 6 solution to a FileMaker Pro 7 solution.

Optimal FileMaker Pro 7 Design Methodologies: An Overview

There are many ways to design a solution in FileMaker Pro 7. There are costs and benefits to each approach to solution architecture. These issues are elaborated upon in *Migration Foundations and Methodologies*. The following overview addresses "Premise # 1" from the introduction: "Learn FileMaker Pro 7 before you convert an existing solution" and the assertion that "if you don't have a good idea of where you're going and how best to get there, you won't end up where you intend to be."

1. **Single file** – Everything is in one file, including schema (tables, fields, relationships), scripts, layouts, access privileges, and data. This allows for portability of the file without losing a file's dependencies. The tables and the relational structure can be seen in one view, and initial design as well as ongoing modifications to the database can be made very efficiently. Security can be managed in one place, with comprehensive and consistent rules easily enforced. A Database Design Report, generated by FileMaker Developer 7, will include information about the solution in one report.

2. **Separate Data and Presentation files** – It is most practical for many solutions to put the interface into a separate file from the data. This is especially beneficial for commercial solutions, or for any solution in which development is expected to continue after the initial deployment (which applies to almost all scenarios in the real world, and is a primary advantage of FileMaker Pro in the first place). To upgrade a file that contains data, it is necessary to either do development on the "live" files, or to import the data into a clone

of the new file. In FileMaker Pro 7, with multiple tables in a file, it is necessary to import all tables of data when performing an upgrade. This can be scripted, but it is a potentially serious constraint for rapid ongoing development of larger solutions.

With some effort and skill, and with certain substantial design constraints, it is possible to eliminate calculations from a data file, and to manage all calculations instead in either another file or in script steps, but it is not practical to do this in most cases. Therefore, to manage updates to solutions it still, at least at times, requires either developing on live files or importing data occasionally.

One common misconception, when developers first discover the ability in FileMaker Pro 7 to put the interface into a separate file from the data, is that it is possible to leverage the relationship graph in the data file when in the interface file. This is not the case. Each file needs to have its own graph if it relies on relationships for any purpose. Relationships are needed in the data file for calculations or lookups. In the interface file, table occurrences that refer to the same base tables, along with a similar relationship, are needed to be used by scripts and for portals on layouts. For that matter, all layouts in a file depend on a table occurrence in the first place, and that table occurrence must be in the same file. Graph elements cannot be copied and pasted from file to file (nor within a file) so it is necessary to manually define a similar graph, or at least a similar "Table Occurrence Group" in more than one file. Nevertheless, despite some inconveniences, there are considerable advantages to separating data and interface.

When you analyze where most time is spent developing a solution, you will likely discover that the great majority of time is spent in layouts and in ScriptMaker, rather than in fields or relationships. Similarly, the ongoing enhancements to a solution over time are largely, though certainly not exclusively, concentrated in these areas. By anticipating schema requirements (tables, fields, relationships) in advance, it is possible to make schema changes relatively rarely, and to replace interface files more frequently. This allows for better quality control for upgrades and less disruption for users.

An important corollary fact which supports this approach is that it will be possible to make schema changes, via the "Define Database" menu item, while files are hosted and in use, bypassing the need for importing data in some situations. Since all schema changes in a given file are committed at one time (rather than incrementally as fields and relationships are changed), reliable use of a solution is not being disrupted by partial schema edits being in place.

3. **Multiple File Solutions** – A common reason to design solutions with multiple files is to support different "data domains". For example, in an enterprise business management situation, separate domains could be defined as: Contact Management, Document Management, Human Resources, Project Management, Purchasing, Billing, Inventory, Accounting, and Financial Analysis. Each of these domains may have several tables of data. The interface to these tables could be in the same file as the data tables, or there could be one or more separate interface files per data file. Alternately, there could be fewer interface files that access multiple data files, or even just one interface file for an entire solution that has multiple data files.

   One of the key benefits to developers is that, when upgrading solutions, the upgrades can be provided in modules, without having to upgrade an entire solution at once. This has benefits for quality control and version control, and will have a potentially big impact on the time or disruption to business operations

required for an upgrade, especially if the upgrade involves files that contain data tables. In a large solution, having many large tables in one file could be problematic because of the time required for importing data just to make changes to a single table's field definitions.

Different interface or presentation files can be created to manage different business processes, yet which reference the same data files. While it is certainly possible to design an interface to behave conditionally based on the user, and to put multiple business processes into a single file, it may be more efficient to maintain different files for different groups of users if their needs or business processes are sufficiently distinct. For example, a "data entry" file can be a separate module from a "reporting" file, yet they reference the exact same data tables in a third file.

A common need in more complex solutions is to manage information about your solution within the solution itself, in "meta-tables", short for "metadata tables". Metadata is data about your database. You may store information about files, tables, layouts, windows, or business processes. You may reference this information only occasionally, such as when upgrading the system or performing certain administrative functions, or it may be a central component of your navigation and system management. Meta-tables could be located in the same file with the interface, or they could be in a dedicated file.

Another motivation for maintaining more than one file is to enable multiple developers to more easily collaborate on development. It is certainly possible for multiple developers to access the same file on a network and do development at the same time, with one substantial limitation. Two developers can open Define Database in the same file concurrently, though only one can make changes at a time. However, only one user can view scripts in ScriptMaker in a given file at a time, which could be very inconvenient for teams of developers collaborating on a solution if all of the scripts are in one file! It is also not as practical to collaborate if you have only one interface file if developers are in different locations unless they have VPN access and a fast reliable connection to a shared server. If the modules of a solution are separated, then disperse teams can more easily collaborate on development.

Modular design also allows certain "components" to be reused in multiple solutions. With important implications for the future of the market, in FileMaker Pro 7 it is much more practical than before for a developer to build a file, or a set of files, and to provide it to other developers for use in their solutions. Access privileges can be configured such that the intellectual property of a developer's product can be protected and also to prevent features from being broken by mistake, while allowing seamless integration into another set of files. With a strategic approach to design, the ease-of-use advantages of FileMaker Pro 7 can be realized by end users or casual developers, while leveraging the unique expertise and prior investments of professional developers. Similarly, professional developers can leverage the more specialized expertise of other developers.

4. **Web Publishing and data exchange with other systems** – Another important consideration when designing or enhancing a solution is to consider the new power of the Internet publishing and data exchange options of FileMaker Pro 7 and FileMaker Server 7 Advanced. It is easier and more powerful than

ever before, thanks to Instant Web Publishing (IWP), Custom Web Publishing (CWP), XML, ODBC, and JDBC, to publish data in a FileMaker Pro database so that it can be accessed in a web browser, and so that it can share data with other systems. See the associated Tech Briefs on Server and Web Publishing for more information.

Assessing and Planning the Migration

Historically, migrating from one version of FileMaker Pro to the next has been relatively straightforward, and those eager for the new power and flexibility of FileMaker Pro 7 may be tempted to "convert and run". While this will undoubtedly be a valuable experiment (and with some solutions, it may actually work), be aware that initial signs of success may be deceiving. The power and flexibility of the new FileMaker Pro 7 architecture comes along with some subtle as well as significant differences, even in pre-existing features. Upon further testing and scrutiny, it will become apparent that subtle differences may produce unintended results.

Assessing Your Solution

It is important to assess the existing solution. This means both assessing the solution from a technical point-of-view and from a business systems perspective. If you are the principal or sole developer of the solution you are converting, and if you have a good memory of what you have done, then some of the following points may not apply to you. However, many developers or system administrators who are converting a FileMaker Pro 6 solution were not the original developer or are not familiar with the details of the solution.

Most importantly, and subtly, sometimes the workflow and business processes that the users of a system depend upon are not known to the developer, because users have "invented" workflow that was not intended, or have creatively adapted to business demands and worked around the limitations of a solution.

If there are serious design problems in your existing solution or if it doesn't effectively address your current business problems or if it contains a lot of obsolete features, then that could be a motivation to rewrite the solution from scratch. Conversely, there may be a large amount of well-conceived organizational knowledge and business systems embodied in your solution, and it is important not to cavalierly underestimate the consequences of rewriting nor, for that matter, of converting without properly attending to the process of pre-conversion and post-conversion modification, testing, and verification.

Premises

1. **Learn FileMaker Pro 7 first** – if you don't have a good idea of where you're going, you're unlikely to end up where you want to be.

2. **Know your solution and your business.** One methodology does not fit all. Conceive a plan for migration that is most appropriate for your situation.

3. **Create multiple stable points** in the migration process where your solution is functional and provides value. In most situations, the migration process involves considerations about enhancing the features of the solution. Unless there are no constraints on time and budget, which is rare in the real world, then you need to make practical decisions – strategizing an incremental approach to achieve an ultimate goal.

Learn FileMaker Pro 7 before converting a solution

Build a simple solution. Explore the example files and tutorials that come with FileMaker Pro 7. Look at solutions that others have written and see how they are designed. Read the other Tech Briefs on Security, Server, and Web Publishing. Consider taking a training from an independent training company or enrolling in the FileMaker Professional Training Series courses. For more information: http://www.filemaker.com/professionaltraining/.

Once you have a general understanding of the product, convert a simple solution, or a subset of a larger solution. Don't try to begin learning FileMaker Pro 7 by working with a conversion of a complex solution. The architecture of a converted multi-file FileMaker Pro 6 solution is quite different than an optimal FileMaker Pro 7 solution; so to start there generates confusion more than benefit.

Familiarize yourself with the Conversion Process and Issues

Read the document that is included with FileMaker Pro 7 and FileMaker Developer 7, entitled "Converting FileMaker Databases from Previous Versions". It contains much useful information. There is also much relevant information in the FileMaker Pro 7 User's Guide, and the other Tech Briefs on Server, Security, and Web Publishing, and the extended documentation on *Migration Foundations and Methodologies,* downloadable from http://www.filemaker.com/upgrade/techbriefs.html.

Interviews

Interviews are a primary means for gathering this information. Conduct interviews with users of each area of a solution. Interview managers and business owners to understand the intent as well as the practical use of the system. If you are not the original developer, then, if at all possible, locate the original developer and interview them to learn as much as you can about the solution architecture, and, if practical, arrange for them to be available to answer specific questions as you encounter issues in the course of migrating the solution.

Develop a Testing Process

Ultimately, the only way to know whether your migration is successful is if it produces the intended behavior. If the objective is to preserve the functions and processes in the current solution, then make a list of those functions and processes for testing at various points in the migration process. Remember that the same process can produce different results under different data and system conditions, so be prepared for iterative and thorough testing.

Analysis and Migration Tools

There are tools available to make it possible to efficiently analyze and report on the existing features of a solution or to automate parts of the migration process, especially helpful in the key phases of fixing specific problems, or when moving a feature to a new file as part of the migration process. These tools are also helpful when you are in the planning phase for assessing the scope of a solution and the potential issues. (See Appendix 2 for more information.)

Define Strategy: What to Convert vs. Rewrite

The most critical decision you can make is regarding what approach you are taking to migration. Use the information in this document and the *Migration Foundations and Methodologies* document to assess what makes the most sense for your situation. There is no more fateful decision than taking an unconsidered approach to migration strategy.

Be willing to have false starts. Trial and error is an inherent part of the process. All of the reading and planning you can do may not provide the experience and perspective gained by working first hand. If you realize part way through the process that you have taken a wrong turn, don't hesitate to go to a backup and start fresh – either from the beginning or from a mid-point. **Save backups from every stage of the process!** It can save a lot of time and money to admit mistakes and start again.

Don't believe everything you read or hear! There are lots of opinions (including in this document) about how to use and take advantage of FileMaker Pro 7, and how to migrate a solution. Build some experience of your own so that you can make the best possible decisions.

Define Priorities and Phases

Based on the interviews or your own knowledge of the business problems that your solution solves or could solve, it is essential that you attempt to accurately recognize the most critical priorities of the system, and make sure that any aspects of it that you intend to rewrite or enhance are truly the most important areas. Relatedly, there may be some problems which are best solved by the features of FileMaker Pro 7 and FileMaker Server 7, so that could be another motivation for prioritizing one area over others. In the course of migration, there may be "down time" in which certain features will need to be offline or not yet implemented in a new or enhanced system. Make sure that you have accurately recognized which features will be least disruptive if unavailable for a period of time.

Hardware and OS Requirements

The hardware and OS requirements of FileMaker Pro 7 and FileMaker Server 7 are different than for previous versions. Notably, Windows versions prior to Windows 2000 are no longer supported, nor is Mac OS 9. FileMaker Server 7 can now take advantage of multiple processors and more RAM, and the hard disk requirements of FileMaker Pro 7 files are greater than in previous versions. More detailed information is available with the products, or on the FileMaker, Inc. web site http://www.filemaker.com.

Estimate Time and Costs, Set Expectations

You probably can not answer a manager's question such as "How big of a project are we talking about?" with "Bigger than a bread basket."

For those who have experienced the ease of upgrading to prior versions of FileMaker Pro, they may underestimate just how different the process is of upgrading to FileMaker Pro 7. It is important that expectations have some connection with reality. You need to allow yourself some time for familiarization and analysis before you can provide a reasonably accurate estimate. So, when your manager asks, "How long will it take and what will it cost?" you can confidently answer, "8 or 17, add your own unit."

A very rough guideline for assessing the time and cost of converting and making a solution work as it did previously, including the process of learning and mastering the issues and learning how to address them, and NOT including adding new features or optimizing the solution architecture to take advantage of the FileMaker Pro 7 architecture, is 10% of the total investment in the product. For example, if development time has been 1,000 hours, then a rough estimate is to expect up to 100 hours will be required to complete the migration. This assumes a comparable level of skill is applied to the migration process and to the learning of the new product as was applied to the prior version. It also assumes that a systematic approach is taken to the migration process, not a haphazard approach.

If not approached strategically, the consequences of database problems resulting in data problems and productivity problems could cost far more in time and money than the investment in a methodical approach to migration.

If the intention is to evolve a solution to a more FileMaker Pro 7-optimized design, then additional time should be anticipated, though there are substantial immediate benefits to converting solutions and just getting them working reliably, as well as advantages which can be realized by designing new features without modifying existing features.

Use a Checklist, with Annotations

There is a useful checklist that is provided with the Conversion Guide included with FileMaker Pro 7 and FileMaker Developer 7. It is strongly advised that the items in this checklist be attended to without exception, though some of the items will not apply to all solutions and may be quickly checked off.

Appendix 1 – Other Important Reference Documents

Upgrading to FileMaker 7: Migration Foundations and Methodologies – available at: http://www.filemaker.com/upgrade/techbriefs.html

Converting Databases to FileMaker Pro 7 (.pdf included with FileMaker Pro 7 and FileMaker Developer 7)

FileMaker Pro 7 User's Guide and *FileMaker Developer 7 User's Guide* (in box)

Other Tech Briefs available at http://www.filemaker.com/upgrade/techbriefs.html

Upgrading to FileMaker 7: How To Take Advantage of the New Server Model and Capabilities

Upgrading to FileMaker 7: How To Employ the New, Advanced Security System

Upgrading to FileMaker 7: How To Benefit From Powerful New Web Publishing Capabilities

Appendix 2 – Analysis and Migration Tools

The **Database Design Report in FileMaker Developer 6** provides an HTML and/or FileMaker Pro database output of most of the metadata in a FileMaker .fp5 solution.

The **Database Design Report in FileMaker Developer 7** provides an HTML or an XML output of the metadata in a FileMaker .fp7 file (but will not show dependencies between files).

MetadataMagic from New Millennium Communications, http://www.newmillennium.com, provides a comprehensive FileMaker Pro .fp5 database of all of the metadata in a FileMaker .fp3 or .fp5 solution.

MetadataMagic also includes **File Reference Fixer**, a tool that prepares file references for conversion to FileMaker Pro 7.

MetadataMagic 2.0 includes a **Conversion Issue Report**, which reports potential problems in solutions that will occur upon conversion to FileMaker Pro 7.

Analyzer from Waves In Motion, http://www.wmotion.com, which takes the DDR output from FileMaker Developer and puts it into a FileMaker Pro database.

Brushfire from Chaparral Software, http://www.chapsoft.com, which provides very rapid and dynamic script tracing in an HTML interface.

Password Administrator from New Millennium Communications, http://www.newmillennium.com, provides a Password Standardizer tool which will enable all passwords in a file to be "case consistent" which is required in FileMaker Pro 7 yet wasn't in FileMaker Pro 6.

Conversion Log Report from New Millennium Communications, http://www.newmillennium.com, imports the Conversion log that is created by FileMaker Pro 7 when a file is converted, and reports any important problems that occurred in a helpful interface.

Table Consolidation Tools – products will soon be available to automate the consolidation of tables and fields from one FileMaker Pro 7 file to another, including bringing over fields with their field types, and at least most of their field definitions (calculations, lookups, etc.).

Access Privilege Schema Planner – given the new power and potential complexity of the access privilege capabilities in FileMaker Pro 7, it will be very useful to leverage a tool for planning this schema in a central interface.

Credits

Danny Mack is the President of New Millennium Communications, Inc., a FileMaker Solutions Alliance Partner based in Boulder, Colorado. New Millennium specializes in FileMaker Pro consulting and solution development, and is the publisher of numerous plug-ins and tools that facilitate the work of FileMaker Pro developers, available at http://www.newmillennium.com.

The real credit goes to the executive management, product management, and, most of all, the engineers at FileMaker, Inc., who have had the vision and the persistence to build an extraordinary technology which enables the creation of true solutions, and will propel the fortunes of many organizations and the livelihoods of developers, subject matter experts and professionals alike, for years to come.