



Demystifying FileMaker Pro File Recovery

Table of Contents

Introduction	3
What are digital assets? What are FileMaker Pro files?	3
What is file damage and how does it occur?	4
Protecting the Integrity and Availability of FileMaker Pro files.	6
Preventing Damage To FileMaker Pro Files	6
Amelioration Of Damage When It Occurs.....	8
What Is The Concept Behind The Recovery Function?.....	10
The Recovery Function In FileMaker Pro 10	14
Reading The Recovery Log	21
Some Common Use Case Scenarios.....	23
What Do You Do With The End Product Of The Recovery Process?.....	24
Summary and Review	27
About The Author	28
Acknowledgements.....	29



Introduction

The great 20th Century statesman and British Prime Minister Sir Winston Churchill once characterized a complex international political issue as

"... a riddle, wrapped in a mystery, inside an enigma...."

"This insightful description fully and accurately describes how most FileMaker Pro developers feel about the File Recovery process, something that has been part of FileMaker Pro for many versions reaching over a decade into the past.

What does it do? Is it safe to use? When does it work correctly? And, perhaps most significantly, is it safe to continue to use a recovered file?

This White Paper is going to attempt to address these and other questions related to the File Recovery process. In doing so, I plan to cover a wide-ranging set of issues and behaviors. In addition, I plan to describe the *new options for file recovery* that are now included in the recently released FileMaker® Pro 10. This White Paper will address six major areas:

1. I want to demystify the purpose, process, and results of File Recovery;
2. I will describe what FileMaker Pro files are and how they are constructed;
3. I will describe what file damage is and how to prevent and to ameliorate it;
4. I will explain the new options for file recovery in FileMaker Pro 10;
5. *I will discuss what are reasonable and prudent assumptions and actions to take based on the results of Recovery; and,*
6. I will provide a description and some heuristics for several use case scenarios for FileMaker Pro file Recovery.

File recovery occurs—just as does file usage—in real-world deployments, not in theoretical or controlled laboratory environments. Nevertheless, there are a number of Best Practices that we can employ both with the development of FileMaker Pro files and with their hosting. In my view, these Best Practices **should be employed**, and the *failure to do so is the major contributor to the file damage that necessitates Recovery's being invoked*. Therefore, the reader should not be surprised that I will stress these Best Practices, all of which came about as the result of some activity or failure that caused problems.

As with every other item dealing with risk management and mitigation of adverse consequences, developers, IT administrators, and business owners must weigh and compare threats, risks, and levels of stringency employed to protect assets. I prefer to take a very careful, cautious, and conservative approach with my clients' databases; and, truth be told, I prefer to do the same with my own.

What are digital assets? What are FileMaker Pro files?

In order to understand file recovery, we must first understand how FileMaker Pro databases are constructed. As developers, we understand the process by which we construct them by employing the features in the *Manage Database* options and elsewhere to create tables, fields, file references, Accounts, Privilege Sets, custom



menu items, value lists, scripts, custom functions, *etc.* However, what have we actually built when we are through doing all this?

FileMaker Pro files are electronic media consisting of multiple instances of 4 KB blocks of information. There are four types of blocks in a FileMaker Pro file:

- The *header block*—one per file that contains information about the version of the file;
- *Data blocks*—many per file that contain the data and structure of the file;
- *Index blocks*—many per file that contain information about the data blocks or about other index blocks; and,
- *Free blocks*—blocks that once held data or index information but that are no longer used.

Damage to one or more of these blocks in any of several different ways is what constitutes *file corruption*. The blocks can lose their references to other blocks; or, the internal order and consistency of a block can be damaged; or, an index can point to something that is not in the file any longer; or, an item that should have an index pointing to it can have lost that reference.

Each block, irrespective of its type or purpose, has certain common characteristics:

1. Each block has a header that indicates what type block it is.
2. Each block points to one prior block and one next block.
3. The internal size of the block may be much less than the 4 KB allocation for a block.
4. Every item in the block has a logical address.
5. Data blocks have starting and ending address.
6. Data blocks are indexed by starting address.
7. Index blocks point to other index blocks or to data blocks.

In addition to these organizing blocks, a FileMaker Pro file also has *Catalogs*. Catalogs are a collection of items that share properties. A catalog may have one or more ordered lists of the collected items, such as a list of the items ordered by item name. An example of this is the *Table Catalog* that has a name list and a custom order list, which is the custom order that the developer has created in the Tables tab of *Manage Database*. Another example is the *Script Catalog* that has a name list, a custom order list, and a quick reference list for those scripts that are listed under the Scripts menu.

Finally, a FileMaker Pro file might have *Libraries* as well. These are data found in container fields of various types ranging from JPG and BMP files to items such as Excel, Word, QuickTime, PDF, or other FileMaker Pro files.

When any of these items incurs damage, problems for the file occur.

What is file damage and how does it occur?

FileMaker Pro files can sustain damage to their structure. The block structure described above can become damaged and no longer be intact. For example, the starting or ending addresses can become damaged. Perhaps the index to a



particular item no longer points to where it should locate that item. Or, FileMaker Pro is unable to determine the next and previous blocks for any given 4 KB block.

A data block can become invalid. In such circumstances, the database engine determines that the block header or structure is damaged in a non-repairable way. Perhaps the logical addresses of items in a block are not in ascending order; or, possibly the sizes of items in block add up to more than 4 KB. Invalid data blocks are detected and discarded in Step 1 of Recovery process as we will subsequently see.

Likewise, one of the database's catalogs can sustain damage. In this instance, for example, there might be items of the catalog's type actually in the database that are not in the catalog's list of items. Conversely, there could be items in the catalog's list of items that are not actually in the database. The Recovery process rebuilds a catalog so its list of items includes only those that actually exist in the database.

What can cause such damage? There are a host of activities and actions that can damage not only FileMaker Pro files but also any electronic asset in any program such as Microsoft® Word, Microsoft® Excel, graphics programs, etc. Hard drive failures, sudden loss of power to the computer, force quitting an application, an OS level or application level crash, two programs trying to access or to lock the same part of the file at the same time (such as an application and an anti-virus program)—all these can damage or corrupt files produced in almost any program. Therefore, *file damage is **not** just a FileMaker Pro phenomenon.*

How can we recognize such damage when we encounter it? The answer might seem obvious, but a closer examination will yield surprising results in many instances. This is part of the mystery of file corruption and recovery. Certainly, there are some directly observable instances of file damage:

- The file will not open and reports itself to be damaged;
- FileMaker Server will not open the file; or,
- FileMaker Server closes the file and reports an error.

Additionally, field index damage can display itself in some telltale fashions:

- Data have been entered into records but cannot be located by Finds; or,
- Data have been entered into records but cannot be located by joins defined in the Relationship Graph.
-

In neither of these instances are the data actually missing from the file; they simply cannot be found by expected methods.

Layouts, scripts, or other similar catalog objects can display damage as well:

- Viewing or printing a layout causes FileMaker Pro to crash;
- Developers are unable to edit a script or a value list; or,
- Scripts, layouts, etc. refuse to change order in their respective list when dragged.

Much to the surprise of some developers, some behaviors **are not the result** of damaged files. Among these we find:



1. Missing data due to poor hosting habits such as peer-to-peer connectivity with multiple copies of files being extant on a network or the exceptionally bad practice of accessing files by OS level file shares on either Windows or Macintosh platforms.
2. A crash caused by a corrupted graphic placed directly on the layout, although the crash *per se* can cause damage to the file.

Protecting the Integrity and Availability of FileMaker Pro files.

Protecting the integrity and availability of FileMaker Pro files is an integral and important part of FileMaker Pro security. In *FileMaker Security: The Book* and in the various Security Tech Briefs found on the FileMaker, Inc. website, I point out that the very essence of security is directed at preservation of the *Confidentiality, Integrity, and Availability* of FileMaker Pro files and the data and business processes they contain. If a database is damaged, certainly the Integrity and the Availability of that file can be compromised, oftentimes in disastrous and irreparable ways.

Effective and professional management of FileMaker Pro databases **requires, not merely suggests**, that rigorous and stringent steps be taken to safeguard the physical integrity of files and to avoid damage to them that would compromise their Integrity or Availability. Likewise, professional management requires stringent and rigorous steps to avoid damage that interferes with or compromises the integrity of business rules and processes that the database is supposed to enforce.

The level of stringency and the degree of rigor employed on a case-by-case basis will be different. The level of effort should also be directly influenced by the level of *adverse impact* on an organization's people, business processes, reputation, assets, and business functionality that damage of one degree or another to a database would precipitate. For example, in the event of an incident what is more important: 100% data recovery or rapid return to uptime? The more damaging the impact of a loss, the more stringent and rigorous the efforts should be to prevent the damage and to mitigate the impact of damage if it does occur. ***This is both elemental and fundamental; but this simple point is often overlooked or ignored.***

Preventing Damage To FileMaker Pro Files

A number of senior members of the FileMaker Pro developer community, together with FileMaker, Inc. System Engineers, Tech Support Managers, Product Mangers, and Development Engineers have consistently over the past seven or eight years argued for and advocated a wide-ranging series of Best Practices to prevent file damage and data loss when using FileMaker Pro files. There have been two White Papers by the FileMaker, Inc. System Engineers, two Server Tech Briefs written by the world's leading expert on the subject, three series of tutorial videos on FileMaker Server and hosted files, a Best Practices White Paper jointly written by a developer and a FileMaker Systems Engineer, presentations by the Lead Server Engineer from FileMaker, Inc. at Developer Conferences, various Webinars by the System Engineers, and, FileMaker, Inc. authorized Professional Training Series classes. All these are directed at describing the known best processes and procedures for professional management of hosted and deployed files and the preservation of the Integrity and Availability of those files.



These recommendations are not the result of some metaphysical epiphany; rather they have come almost exclusively as the result of knowledge gained and tribulations suffered by trying to repair damage to a client or FileMaker customer's databases. I offer them at length here in the hope that they will be useful. **Because the best way to manage file Recovery is to avoid incurring the damage that requires Recovery to be invoked.**

Protection of FileMaker Pro files requires following a *rigorous regimen, not a casual one*. A number of specific recommendations will follow hereon, but at the outset, this much seems clear to me. Developers and Administrators gamble with the integrity and availability of their FileMaker Pro database files if they do not follow Best Practices to prevent file damage. They will lose that wager, more likely sooner rather than later, incurring varying degrees of pain and distress as a result. So, what are some of these Best Practices to protect files against damage?

1. Use FileMaker Server correctly, including appropriate hardware correctly configured, appropriate Operating Systems correctly configured, with correct backups strategies and backup systems in place, and with regular *verifications* of all backups.
2. Use Server appropriate hardware; with the hard drive subsystem being the most important element followed by adequate RAM and adequate processor power. Dual power supplies with UPS units properly and correctly configured are very important. Power outages cause damage to files. The purpose of the UPS is *not* to keep the Server in operation, *but* to keep it running long enough to close the files and stop the FileMaker Server service/daemon so the files do not sustain damage.
3. Properly configure the Operating System, specifically including *disabling* unnecessary, unneeded, and potentially harmful services/daemons.
4. Prevent interference with hosted databases by improper access means such as the Windows Explorer, FTP clients, *etc*.
5. Avoid the introduction of OS level file shares of FileMaker Pro files or of access to such files by OS level file sharing.
6. Avoid having anti-virus programs or disk indexing programs being allowed to access hosted or backup FileMaker Pro files.
7. Avoid automatic installation of updates either to applications or to the Operating System, especially when there is an automatic restart.
8. Avoid defragmenting hard drives with FileMaker Pro files on them. A failed defragmentation can severely damage a file. Remove the FileMaker Pro files, run the defragmentation, and then replace the files.
9. Implement and **test**—repeat **test**—effective backup strategies. The purpose of a backup is **not** to have a backup of the file. The purpose of a backup is to be able **to restore the file** if something happens to it. Untested backups are probably worse than no backups at all, because they impart a false sense of security and well being. The time to discover that your backups cannot restore the file is **not in the immediate wake** of an incident that has damaged the file. **A rigorous backup regimen is an indispensable pre-condition for being able to utilize various components of the FileMaker Pro Recovery process as we will see in later sections of this White Paper. If you do not have good backups, your ability to fine**



tune the Recovery process will be *seriously constrained and limited.*

10. Avoid allowing third-party backup tools to access hosted FileMaker Pro files. This includes such items as Retrospect, BackUp-Exec, and Time Machine. These systems should *only* work on copies of backups, never on the live files, and prudently not on the first, local backup.
11. Utilize FileMaker Server's backup verification to test the soundness of your backups. *Read the logs for any information about errors or failures of verification.*

In addition to a rigorous regimen for hosting and deploying files, developers can and should deploy similar precautions while developing files. I would strongly recommend having as *pristine and controlled a development environment* as possible consistent with sufficient risk management. I would employ UPS units on development machines along with *very frequent* backups. Additionally I would eschew all development on hosted files when users are connected. This can lead to loss of structural data in the files as well as induce locking of tables. Additionally, if a user experiences such a lock, he or she may decide to force quit the running FileMaker Pro application and thereby damage any *local* files on that user's workstation.

I fully realize that a prohibition against developing on hosted files can create real workflow obstacles in organizations and that it can be particularly vexing to multi-developer projects. Prudent risk analysis with a *full understanding of potential downside damage* is necessary in these situations. In my view, it is probably best altogether to avoid developing on hosted files if possible. **In any event, never continue to do development work on damaged files.**

To summarize what we have talked about already:

- The best way to prevent having to repair or to replace files *is to avoid having damage in the first place.* Follow Best Practices for both development and deployment to help avoid incurring file damage.
- Additionally, have a contingency plan in place if damage occurs.
- Have a rigorous backup strategy including clean clones of the structure of all files and verify the integrity and the validity of backups. **A perfect backup of a corrupted database is a perfectly corrupted backup.**

Amelioration Of Damage When It Occurs

Sometimes despite all best efforts, damage occurs to files. What can developers and administrators do when this happens? We obviously want to detect the type, nature, and level of any damage and take whatever corrective action we can. An important question I want to try to answer is this: *What is the resulting product after we attempt to repair damage, and what can we safely and prudently do with that resulting product?*

In my view, Recovery of a file should be the last resort. I would recommend reverting to the last known good backup as the first step. That does mean that data can be lost; it also means that schema and structural changes made since that last known good backup are also lost. The data (*i.e.* the user-entered data) can be retrieved and brought into the good file. However, this does not address the issue of



deletes that may have occurred. Updating the total found record set, including deletes, edits, and adds, requires a different process: namely the *roll-forward* of the last known good structural backup or clone.

Roll-forward is not an inherent FileMaker Pro capability. However, third-party tools such as WorldSync's **fmDataGuard** can accomplish this roll-forward functionality. It is beyond the scope of this White Paper to discuss roll-forward's use. However, the basic concept is that roll-forward takes a backup and *updates* it.

So, if we want to work on the damaged file, either to extract data or possibly to extract structural elements, what do we do? Our very first step should be to run a *Consistency Check* on the suspect file. What is a *Consistency Check* and what does it do? Is there anything new about the Consistency Check in FileMaker Pro 10?

First, both FileMaker Pro 10 and FileMaker® Server 10 will run a Consistency Check when a file was not properly closed the last time it was opened and modified. Additionally, this check will also be run the first time a file is opened in either FileMaker Pro or FileMaker Server versions 8 through 10, if the last time the file was opened and modified was in either FileMaker Pro 7 or FileMaker Server 7.

A Consistency Check reports on the state of the file. The Consistency Check in FileMaker Pro 10 will be somewhat more rigorous than the one in FileMaker® Pro 9 in that it will check that all of the logical addresses within the block are in ascending order.

When is a Consistency Check called or run? What does it do? The check will **not** be run in either FileMaker Pro 10 or FileMaker Server 10 the first time the database is opened except as noted above. However, the *Verify Backup* option in FileMaker Server 10 is the same as the Consistency Check. *New in FileMaker Server 10 is the ability to run a scheduled verification on files.* This closes the files, runs the verification, and generates a report to the Server Event Log. **Also new in FileMaker Pro 10 is the capability to call a Consistency Check on demand on a closed file.** I will describe this at greater length later in this White Paper.

The Consistency Check examines only the block structure of the file:

1. It checks previous–next links between blocks;
2. It checks internal size and header of each block;
3. It verifies that the starting address of a data block matches the index pointing to it;
4. It verifies that all address of a data block are in order;
5. It verifies that all blocks pointed to by an index block actually exist; but,
6. It does NOT check record data or schema.

In FileMaker Pro 10, the Consistency Check will present a report of its results. From that report, developers and administrators can make some preliminary decisions about what to do with the suspect files.



What Is The Concept Behind The Recovery Function?

So, once we determine that the file has some damage, how do we fix or repair it? In addition, is the resulting fixed or repaired file then safe to use? The latter question especially is one of paramount importance and concomitant unparalleled uncertainty and confusion.

First, the purpose of the Recovery function is to correct and repair the damage to the extent possible so that as much of the data can be extracted from the damaged file as possible. The process will sacrifice structural elements of the file to accomplish this. *It will also sacrifice offending data elements as well.* This process does not guarantee the production or creation of a useable file. There have been some refinements to this in FileMaker Pro 10 that can make it possible to extract certain structural elements from a recovered file and then put them into a known good copy of that file.

What does the Recovery function actually do? There are at least four major functions that occur in the *default* Recovery process:

1. Recovery scans for valid data blocks in the damaged file;
2. Recovery then creates a new file using **only** the valid data blocks, completely ignoring any index blocks. Only valid blocks are written to the new file. Invalid blocks are dropped and holes patched as if the block were never there;
3. Recovery Scans and rebuilds record and field data of all tables, schema such as field, table, and relationship definitions, and structural items such as layouts, custom menus, and scripts. Recovery *does not repair any data corruption*. Recover may discard—for fields other than container fields—any corrupted data (compressed Unicode that cannot be uncompressed); and,
4. Recovery rebuilds the index for every field that has an index.

There are new options in FileMaker Pro 10 that will allow developers and administrators to exert some fine-level control over each of these items. To understand these options and to make correct decisions about their proper use requires an understanding of how the Recovery process works on a damaged file. So, we will now examine that question. Then, in a following section of this White Paper, we will discuss in detail the nature and best use of these options.

This is how Recovery works. There are sixteen specific steps. In addition, there are Alternate Steps 1 and 2, new to FileMaker Pro 10 that can be employed in lieu of the default ones.

1. **Scan the entire file**, ignoring index blocks, free blocks, and invalid (malformed) data blocks. This process returns an ordered list of all valid data blocks.
2. **Create a new file** by copying valid data blocks to it. Create a new header block and new index blocks. This file would now pass a Consistency Check.

There are **Alternate Steps 1 and 2:**



This is a new feature in FileMaker Pro 10. *Alternate Step 1* is a straight copy of original file as-is with no checking of any blocks. *Alternate Step 2* optimizes the logical structure of the new file such that each item that is stored at a logical address in the old file is individually read from the original file and put into the new file. As a result, the new file will pass the Consistency Check, but *it may miss data that could not be read by logical address because of problems in original file*. Why use these alternate steps? First, it is faster; the speed of Recovery process is enhanced. Second, it offers a fine level of control over desired results from the process as we will see subsequently.

3. **Recover the Font List** such that all fonts that are used in the file are assigned a number and all uses of that font are referenced by that number.
4. **Recover the Catalog of base tables and the Catalog of table aliases** that are constructed on the Relationship Graph. This step does not address the fields that are in each table. Lists associated with these catalogs are usually deleted and reconstructed. However, custom order lists are repaired rather than reconstructed *since deleting them would lose the order*.
5. **Recover the Catalog of Relationships** as defined in the Relationships Graph. As above, lists associated with these catalogs are usually deleted and reconstructed. However custom order lists are repaired rather than reconstructed *since deleting them would lose the order*.
6. **Scan all records and fields for each table to recover the Field Catalog and Library Catalog** for each table. As part of this step, check field definitions, scan data for any issues, and delete and rebuild word and value indices. Libraries contain the items stored in container fields such as graphics, Microsoft® Word documents, or other FileMaker Pro files. There is also a library for the entire file that contains any graphics stored on the layouts.
7. **Rebuild Custom Functions Catalog** including any common catalog-wide processing plus catalog-specific processing for each item stored in the catalog. As part of this step, count items and rebuild the index of ID's, rebuild any custom order lists, remove entries to members that do not exist, compact the entries to remove any holes, and add any missing entries to the end of the list.

The same process can apply to other Catalogs, specifically as noted in Steps 11, 12, 14, 15, and 16, and more generally to all Catalogs found in the file.

8. **Rebuild user Accounts** by rebuilding their lists. The list of Account names and ID's in the Accounts catalog can be rebuilt along with any custom order lists. Some of the hashed information cannot be rebuilt or changed however.

A hash is the one-way, non-reversible result of performing a mathematical rule on a string of data. Even if the hash were recovered, it is computationally infeasible to reverse the process and



thereby to obtain the original data: the password. When the user presents his or her credentials for authentication, FileMaker Pro hashes the credentials and compares them with the ones in the file. If there is a match, the user is authenticated as valid. See Using FileMaker Pro 9—How to employ the New; Advanced Security System at page 23. Also see FileMaker Security: The Book, pp 29-33 [<http://www.filemakersecurity.com>] for specifics about the FileMaker Pro password hashing mechanism.

9. Rebuild Privilege Sets.
10. Rebuild Extended Privileges.
11. Recover File References and Data Sources Catalogs in the same manner as Step 7.
12. **Recover Value Lists Catalog** in the same manner as Step 7.
13. Recover layouts including references to any container data on a given layout.
14. **Recover Scripts Catalog** in the same manner as Step 7.
15. **Recover Custom Menus Catalog** in the same manner as Step 7.
16. Recover the two Custom Menu Sets Catalogs in the same manner as Step 7.



These steps are shown in Chart 1 below.

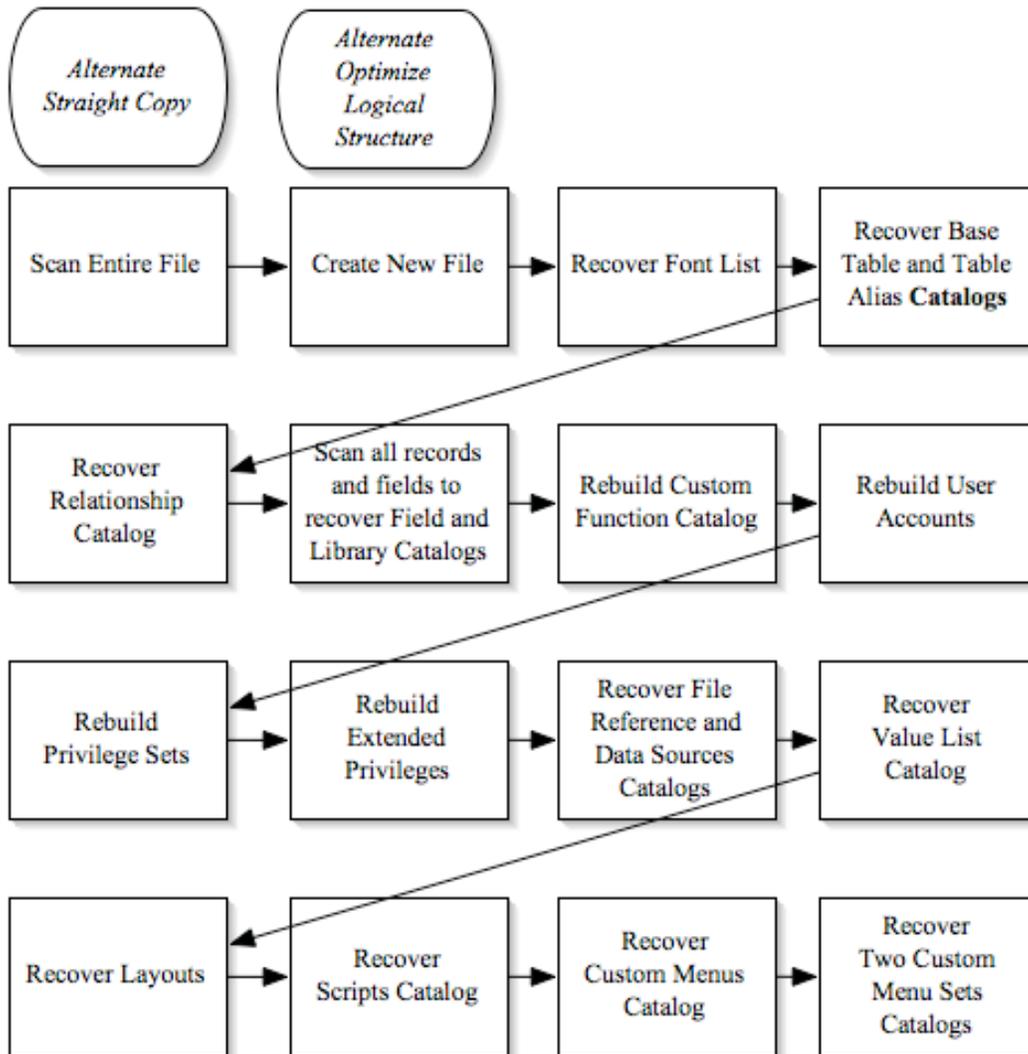


Chart 1: Sixteen Steps In File Recovery and Two Alternative Steps

When these sixteen steps are completed, the resulting recovered file is then available. Its usability and functionality will vary widely, depending in large degree on the level and type of damage to the original file. Thus, the answer to the original—and most pressing—question of whether the file can continue to be used, may well be reflected in the famous response of a character in John Le Carré’s epic espionage novel, *The Little Drummer Girl*:

Yes, repeat, No.

What does this mean? As with all issues related to the Confidentiality, Integrity, and Availability of FileMaker Pro files, developers and administrators must know the *vulnerabilities*, identify the *threat agents* that can trigger *threats* that exploit those vulnerabilities, and determine the *risks* that those threats will actually materialize.



In order to do these, they must also have some knowledge of the severity of the level of *adverse impact* of a breach will be on the organization's people, operations, assets, and reputation.

Thus, at the end of any recovery process, whether full (default) recovery or partial (customized) recovery, the **benefits** of utilizing all or part of the recovered product must be balanced against the risk that use of those parts will trigger some other problem in the future that has an **unacceptable level of adverse consequences on the database file and hence on the organization's people, operations, assets, or reputation**. This is a case-by-case determination.

The Recovery Function In FileMaker Pro 10

The Recovery Function and the Consistency Check remain in FileMaker Pro 10 much as they did in FileMaker Pro 9, but with some useful and important modifications and additions. In this section, we will explore these.

A new feature in FileMaker Pro 10's Recovery options is the ability to bypass a start-up script or layout that might be causing a file to crash. A frequent cause of this unwanted behavior is a corrupted graphic on the splash screen or an opening script stuck in an infinite loop. If a developer or administrator has a [Full Access] account for the file, he or she can invoke this option. This will bypass the start-up script and take the developer to a newly created blank layout.

A particularly compelling new feature in FileMaker Pro 10 is the ability to order a Consistency Check **on a closed file on demand**. After the check is completed, results are reported back by FileMaker Pro. How does that work? Developers and Administrators can call the Consistency Check *via* the Recovery option in the File menu as shown in Figure 1 below:



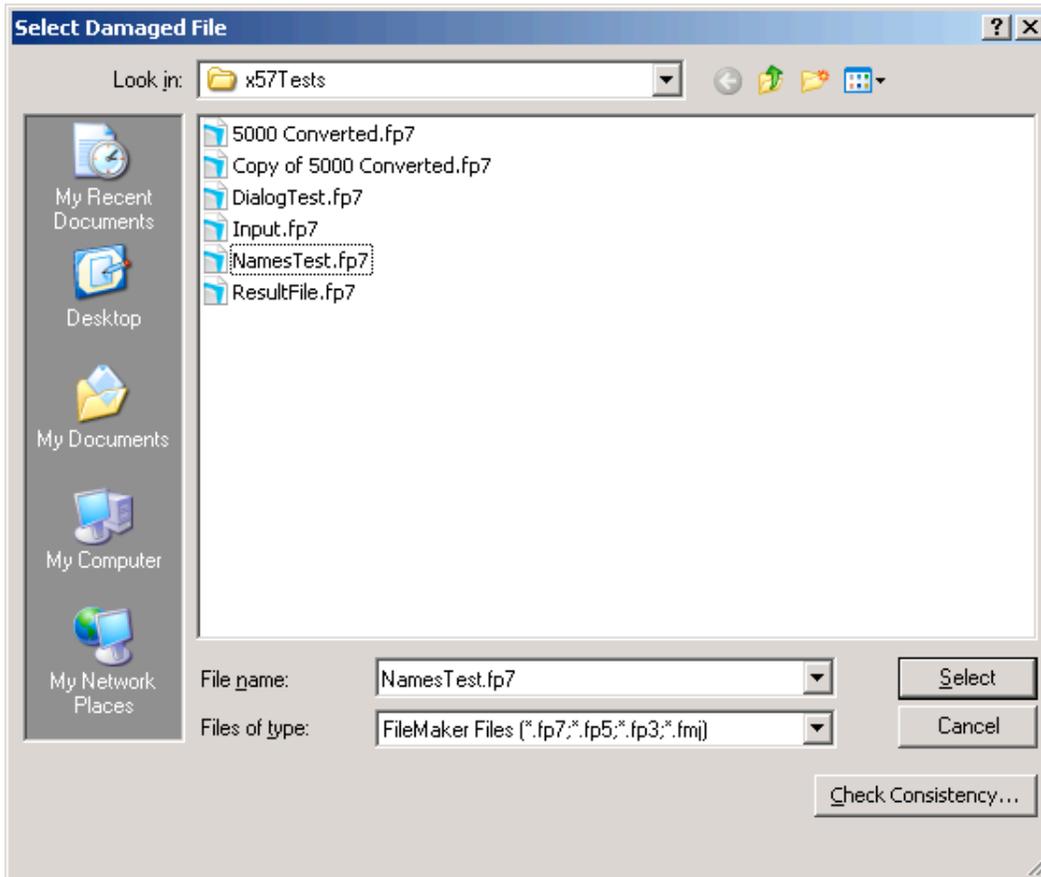


Figure 1: Call A Consistency Check On Demand By Selecting A File And Clicking The "Check Consistency..." button

Depending on the size and complexity of the file, this can take some time to complete, as shown in Figure 2. When the process has been completed, two things will happen. First, FileMaker Pro 10 will return a report as shown below in Figure 3 or Figure 4. Second, FileMaker Pro will write the results to a log. **Pay careful attention to that log.**



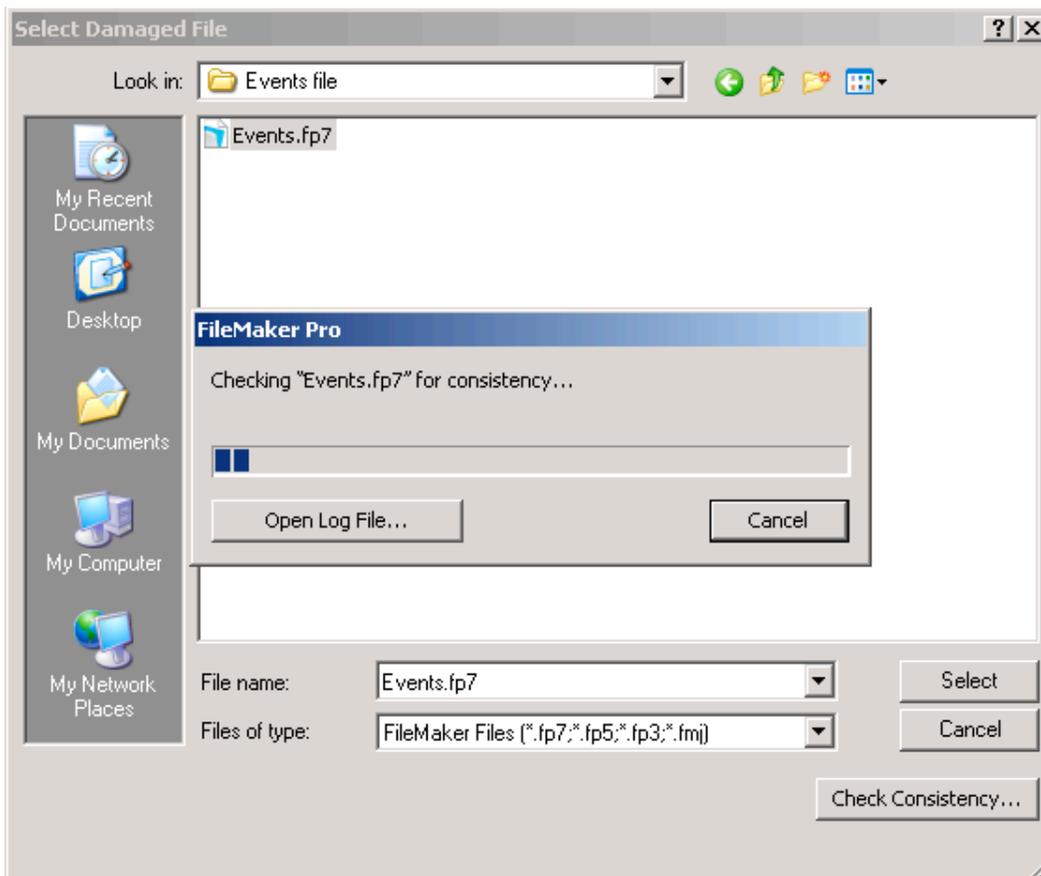


Figure 2: A larger, more complex file needs more time for a Consistency Check.

Figure 3, below, shows the report when the Consistency Check finds no issues. Figure 4 shows the report when there are problems.

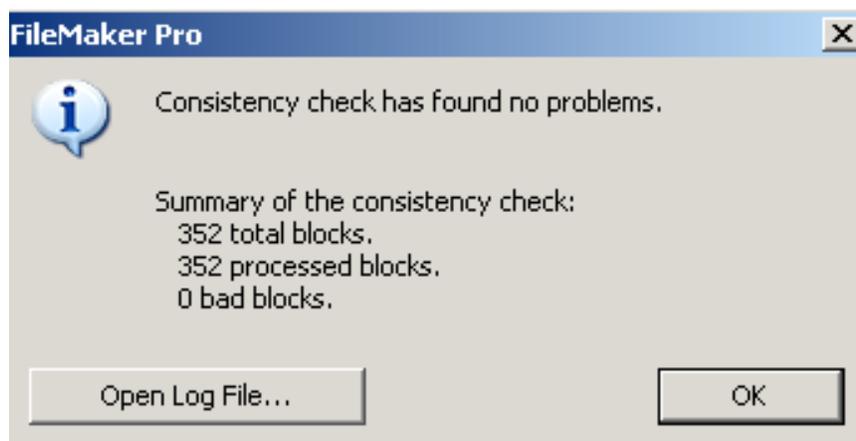


Figure 3: No problems reported by the Consistency Check



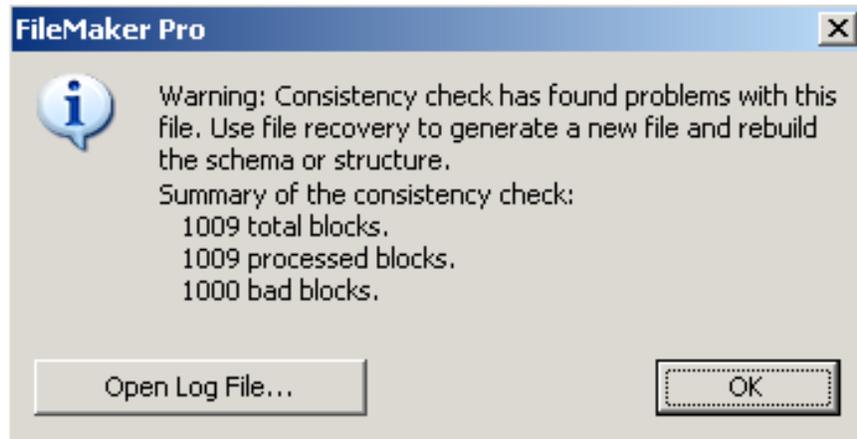


Figure 4: Problems reported by the Consistency Check

FileMaker Pro 10 will also write out information about the results of Consistency Check to a Recover Log created in the same folder as the databases. If such a log previously existed, the new information is appended to the original log. An example of Consistency Check entries in the log:

TIMESTAMP	FILENAME	ERROR	ACTIVITY
2008-10-19 10:53:41.041 - 0500	testOne.fp7	0	*** Started consistency check, total of 1009 block(s) to check
2008-10-19 10:53:41.132 - 0500	testOne.fp7		ERROR: Index does not lead to first entry of page 6; goes instead to page 1005
2008-10-19 10:53:41.142 - 0500	testOne.fp7		ERROR: Index does not lead to first entry of page 7; goes instead to page 1006
2008-10-19 10:53:41.162 - 0500	testOne.fp7		ERROR: Index does not lead to first entry of page 8; goes instead to page 1006

Table 1: Consistency Check Report From Recover Log

Referring to the table above, the last number in the time stamp is the offset from GMT of the computer's clock. The error number column will be filled in only if there is a FileMaker Pro error code that matches the event.



FileMaker Pro 10 allows developers and administrators to select from a variety of options when invoking Recovery. After first selecting *Recover...* from the *File* menu, developers can name the recovered file and select *Use advanced options* as shown in Figure 5 below by clicking the *Specify* button.

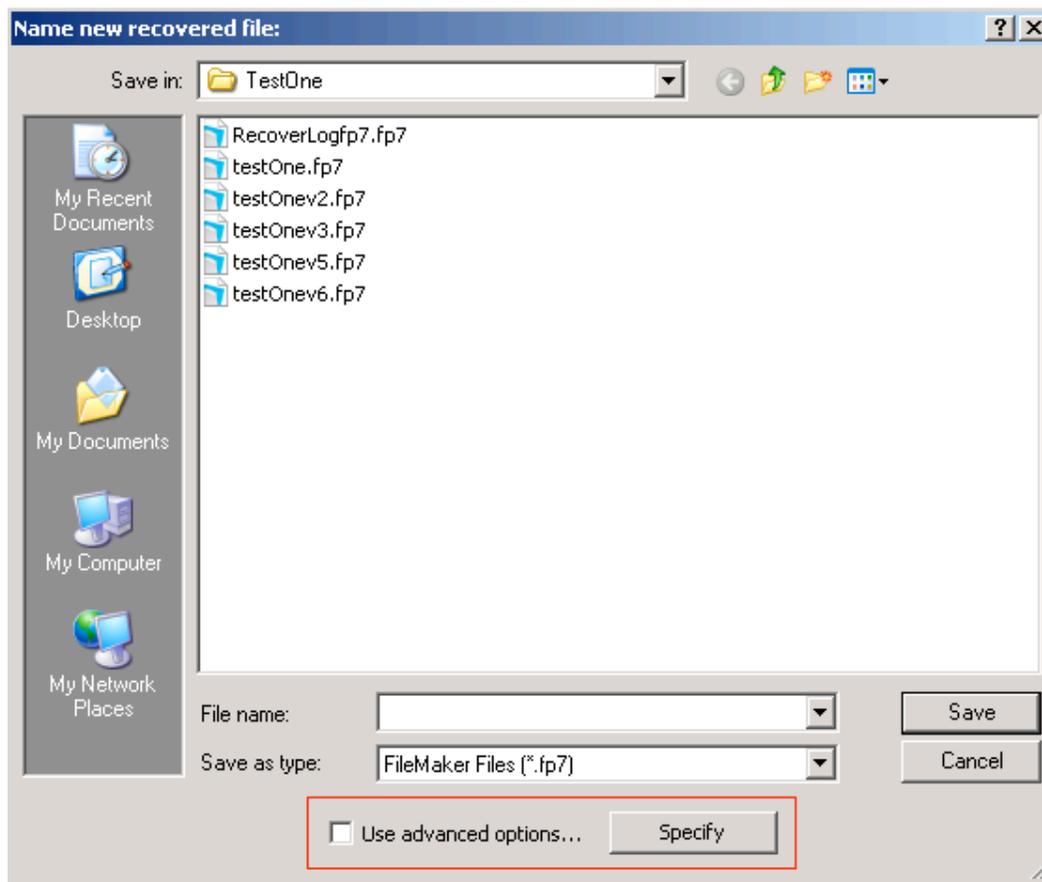


Figure 5: Select the Use advanced options by clicking the Specify button

This in turn will produce the *Advanced Recovery Options* window shown in Figure 6 from which developers and administrators can make a wide variety of selections. **When the Advanced Options are disabled or not selected, Recovery defaults to the same sixteen steps that were described earlier.** *This is important to remember* as is the converse situation. If the developer selects the granular options instead of the default ones, the result of the Recovery may be different than in the default instance. *This is also important to remember.*



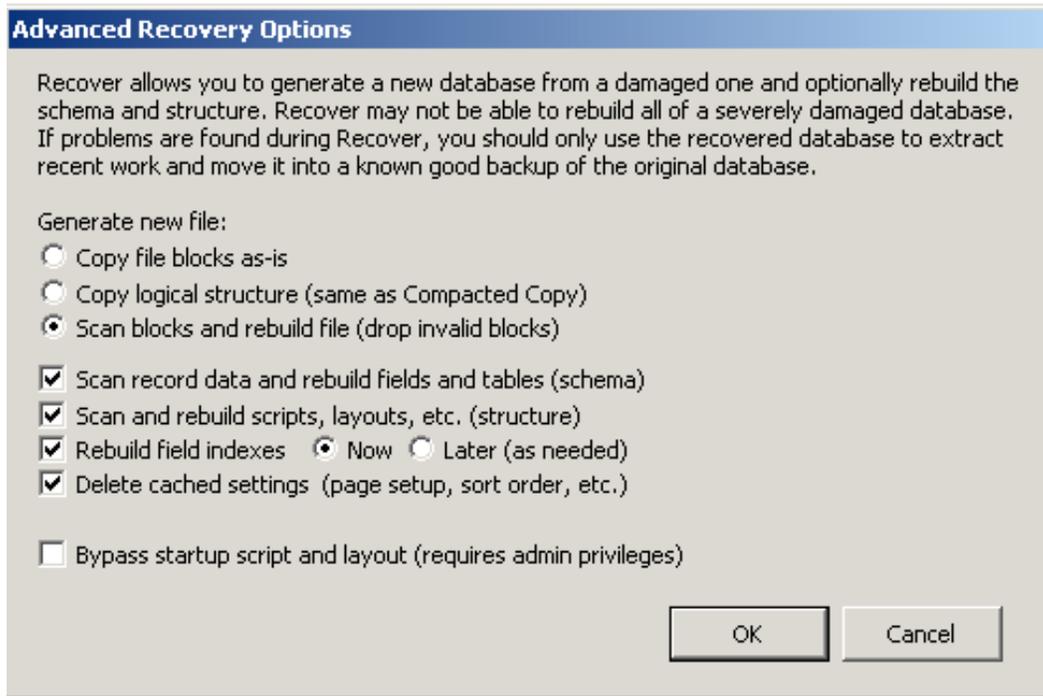


Figure 6: These are the Advanced Recovery Options for both high-level and fine-level actions. Note the individual high-level and fine-level options selected by default initially when you specify use of Advanced Options.

There are three principal *high-level* options for generating the new file:

1. **Copy file blocks as-is:**
 - a. Perform no repairs to any blocks
 - b. Allows advanced developer to recover just the structure (scripts and layouts) or to rebuild field indices.
2. **Copy logical structure (same as Compacted Copy):**
 - a. Uses the logical address structure to copy the **content** of the damaged file to a new file creating all new block structures.
 - b. Has the same result as Save a Compacted copy except that is done **on a closed file**
3. **Scan blocks and rebuild (drop invalid blocks):**
 - a. Exactly the same process as Steps 1 and 2 of default ("regular") Recovery, namely **scan the entire file**, ignoring index blocks, free blocks, and invalid (malformed) data blocks and then **create a new file** by copying valid data blocks to it.

Selecting Option 3, *Scan blocks and rebuild the file (drop invalid blocks)*, on completion will produce this report from FileMaker Pro 10:



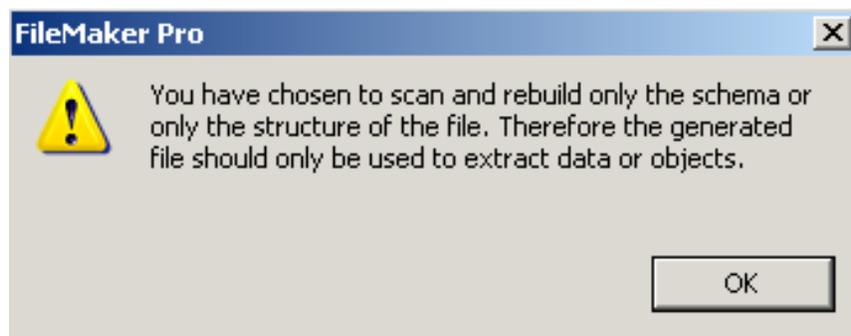


Figure 7: Warning message related to schema or structure

In addition to the three high-level options, developers and administrators can select from a variety of *fine-level*, more granular options. *None*, *one*, or *more* than one of these fine-level options can be selected for whatever high-level option is selected:

1. **Scan record data and rebuild fields and tables (schema)**
2. **Scan and rebuild scripts and layouts, etc. (structure)**
3. **Rebuild field indexes** (can be selected or not, and if selected, has options)
 - a. Now
 - b. Later (as needed)
4. **Delete cached settings (page setup, sort order, etc.)**
5. **Bypass startup script and layout** is a new option for FileMaker Pro 10 and requires a [Full Access] Account.

What do these various options actually do?

- **Scan record data and rebuild fields and tables** is the *only* option which causes *all* of the record data to be read, which may find fields that need definitions created, invalid repetitions, or invalid field data. It will also rebuild the table, table alias, and relation catalogs and rebuild all dependencies (field and relational).
- **Scan and rebuild structure** is the option that causes all of the other catalogs to be rebuilt (scripts, layouts, *etc.*)
- **Rebuild field indexes** offers three options for dealing with field indices:
 - *Disabled* does nothing and leaves existing field indices untouched (and is the fastest Recovery);
 - *Enabled/Now* rebuilds all indices during the Recovery (and is the slowest Recovery); and,
 - *Enabled/Later* deletes existing indices during Recovery and sets fields that had indices to "build indexes automatically as needed" (and is a middle speed Recovery and will slow down some Finds later, temporarily).
- **Delete cached settings** removes a small number of "last used" settings that are stored in the database for the user's convenience (*not* any settings that are stored in scripts) and includes such items as the



last Page Setup dialog settings, the last order used in the Sort dialog, etc. There should be no speed difference in Recovery from this setting.

- **Bypass startup script and layout** allows an administrator with a [Full Access] Account to disable the On Open script and set the opening layout to a new blank layout. This avoids the invoking of an On Open script that has an infinite loop for example or an opening layout with a corrupted graphic, either of which can cause a crash. There are other ways to bypass an On Open script, but this provides an immediate and safe way to do so.

Thus, the developer or administrator is able to construct a very finely tuned set of behaviors for a customized Recovery process. However, he or she should bear in mind that there are *many, many* combinations of different results that can occur depending on what particular set of both high-level and fine-level options the developer or administrator instructs FileMaker Pro to employ.

Reading The Recovery Log

The Recovery process generates an extensive log. **Developers and administrators should consult this log very carefully for additional information about the results of the process.** One of the best features of the log is that it generated in a tab-delimited format. Therefore, developers can easily import it directly into FileMaker Pro for review and analysis. Here is a sample of entries from the log with the timestamp, file name, and error code information removed, and only the *Activity portion* displayed. Note certain lines highlighted in **bold** or *italic*. This table from the log has been purposefully truncated for example purposes. The complete log is approximately 36 pages long.

ERROR: Page 913 contains duplicate data, dropping it
Dropped page starts with data of type: table (132) record data
ERROR: Page 914 contains duplicate data, dropping it
Dropped page starts with data of type: table (132) record data
ERROR: Page 915 contains duplicate data, dropping it
Dropped page starts with data of type: table (132) record data
ERROR: Page 1003 contains duplicate data, dropping it
Dropped page starts with data of type: table (132) record data
Rebuild processed 10 page(s), completing end of file
Completed copy or rebuild of file blocks
<i>Recovering: font catalog</i>
Rebuilding font catalog order list: name
Recovering: font 'Verdana' (0)
Recovering: font 'Lucida Grande' (1)
<i>Recovering schema</i>
<i>Recovering schema catalogs</i>
<i>Recovering: table catalog</i>
<i>Rebuilding table catalog order list: name</i>



<i>Rebuilding table catalog order list: custom</i>
Recovering: table 'Members' (130)
Recovering: table 'Companies' (131)
Recovering: table '_350000' (132)
Recovering: table 'Recovered Library' (133)
Recovering: table alias catalog
Rebuilding table alias catalog order list: name
Rebuilding table alias catalog order list: custom
Deleted relational dependencies
Recovering: table alias 'Members' (1065090)
Recovering: table alias 'Companies' (1065091)
Recovering: table alias '_350000' (1065092)
Recovering: table alias 'Recovered Library' (1065093)
<i>Recovering: relation catalog</i>
Recovering relation from 'Companies' to 'Members'
Recovering fields for table 'Members'
<i>Recovering: field catalog</i>
Rebuilding field catalog order list: name
Rebuilding field catalog order list: custom
Rebuilding field catalog order list: type
Recovering: field 'Member Number' (1)
Recovering: field 'First Name' (2)
Recovering: field 'Last Name' (3)
Recovering: field 'Home Address 1' (4)
<i>Deleted last used page format</i>
<i>Deleted last used print settings</i>
<i>Deleted last used layout formatting defaults</i>
<i>Deleted last used import orders</i>
<i>Deleted last used export orders</i>
<i>Deleted last used sort orders</i>
<i>Deleted last used request sets</i>
<i>Deleted last used found sets</i>
WARNING: problems were detected while recovering the database. The recovered file should NOT be used going forward; copy only the most recent work from it into a backup copy of the original file.
File blocks: scanned 810 block(s) and rebuilt 10 block(s), dropped 997 invalid data block(s)
blocks with invalid structure: 0
<i>blocks with zeroed header: 197</i>
<i>blocks with overlapping data: 800</i>
blocks with unloadable header: 0
Schema: scanned fields and tables; no problems found
Structure: scanned; 0 item(s) modified
File size after recovery is 32768 bytes



*** Completed recovery to 'testOne Recovered.fp7'

Table 2: Activity Portion of Report From Recover Log. Note highlighted areas.

Some Common Use Case Scenarios

There are any number of commonly encountered scenarios where developers and administrators may need to employ the various features of the Recovery process. In this section, I will discuss several of these.

Common scenario #1. FileMaker Server attempts to open the file, then closes it, and the Server Log calls for Recovery or notes that file has failed its Consistency Check.

This is an indication that the file has failed its Consistency Check, although there is no immediate indication of the extent or severity of any damage. As a first step, disconnect any users accessing files on FileMaker Server, then close all files, and then finally stop the FileMaker Server service or daemon. From within FileMaker Pro select the **closed** file that is suspect, and run the Consistency Check on it as described in Figure 1. Examine the reports of the Consistency Check as reported in the UI (Figure 3 and Figure 4) and as written to the Log. In many instances, some simple work on the file will make it able to be opened and used. Note the various high-level options described in Figure 6.

The second option is to copy the logical structure of the file. Utilize that option to produce a new copy of the suspect file. You will recall what happens here is that FileMaker Pro uses the logical address structure to copy the **content** of the damaged file to a new file creating all new block structures. Examine the report of this action, especially checking to see if any blocks were dropped in the process. Run a Consistency Check on that new copy. In many instances, that new copy can then be opened by FileMaker Server. This is the same result returned by the process of saving a compacted copy that was employed in earlier versions of FileMaker Pro. Note that this is not the same process found in earlier versions of using FileMaker Pro Advanced to compact a file. **That process should never be used on a suspect file. In addition, in FileMaker Pro 10, these "file maintenance" options have been removed.**

Common scenario #2. Standalone file will not open and reports itself damaged.

This is an indication of significant damage to the file. I would recommend the following process. First, run the Consistency Check and review the report. Developers and administrators can then try the default Recovery process. Alternatively as shown in Figure 6, you may wish to select the third high-level option to scan blocks and rebuild file, along with several of the fine-level options such as scan record data and rebuild fields and tables while perhaps skipping the rebuild of the file structure. Extract data from the rebuilt file and put it into a known good copy.

Common scenario #3. Records are entered into database, but searches on known field values fail to find them or they do not appear in sets of related records when they should.

This situation usually indicates damage to the file's indices. First, run the Consistency check and review the reports. You then have the option, for example, to



scan and rebuild (high-level) combined with scan record data and rebuild fields and tables along with rebuild field indexes (both fine-level options). If this fixes the issue and if there are no other reports of damage, weigh the risks associated with doing a complete recovery and using the file against the benefits of sending the data to a known good copy. In the past, the recommendation for index repair was to remove indexing from the field, close the file, reopen the file, and then reinstitute the indexing. The new option in FileMaker Pro 10 may offer better results.

Common scenario # 4. Individual scripts in ScriptMaker cannot be edited nor can their ordinal positions in the list of scripts be changed by dragging them to new positions.

This indicates damage to the Scripts Catalog. Run the Consistency Check and review the reports. Here you have an option to try to repair the damage and then copy the Scripts back to the original file. Or, if that is too risky, you could send the data to a known good copy. If selecting the former option, you select the high-level option to scan and rebuild and the fine-level option to scan and rebuild scripts, layouts, etc. (the file structure).

Therefore, there are a lot of options and a great deal of nuance to this process. How are developers to know what to select and what is safe to use? Let us now turn to a discussion of these items.

What Do You Do With The End Product Of The Recovery Process?

At the end of the Recovery process, there are a number of results that FileMaker Pro 10 might report. In the first instance, FileMaker Pro may report that the file was so damaged that no Recovery could occur:

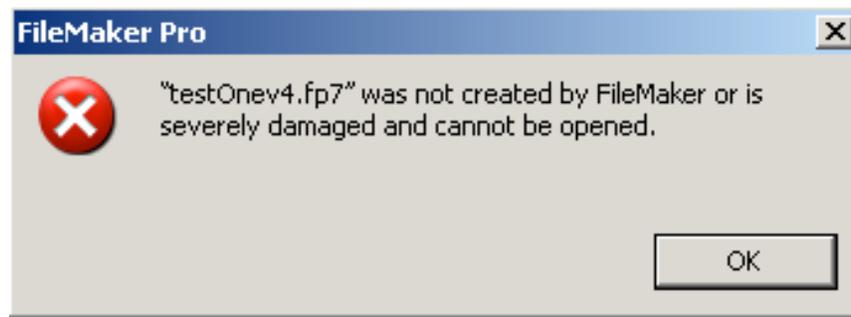


Figure 8: Recovery Failed

In another instance, FileMaker Pro may successfully recover the file and its contents.



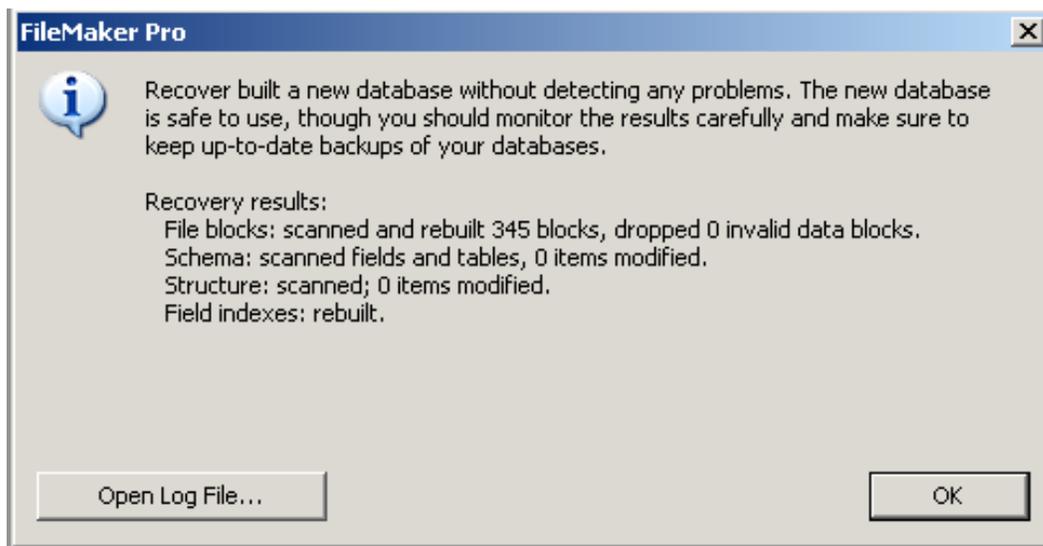


Figure 9: Recovery Succeeded: Note that no data blocks were dropped

Please note the portion of Figure 9 that discusses the number of invalid data blocks that were dropped. In this instance, the message indicates that Recovery "...dropped 0 invalid data blocks." There is a situation where the message could indicate, as it does in Figure 9, that "Recover built a new database without detecting any problems." But the message could also indicate that one or more invalid data blocks were dropped. In such instances, developers and administrators should also consult the Recover log generated by FileMaker Pro. The log will have generated a cautionary warning, but the UI of the message will **not** have that warning. **The dialog presents the warning if there were record or field issues, but does not present a warning if the only problem encountered was block damage. The log is governing here, not the UI with the message.** As previously mentioned, it is always a good idea to consult the log.

In yet other instances, as shown in Figure 10 below, FileMaker Pro, may report that while it was able to build a new database, that problems remain with the file as rebuilt, and therefore, developers should copy the most recent work into a known good backup.



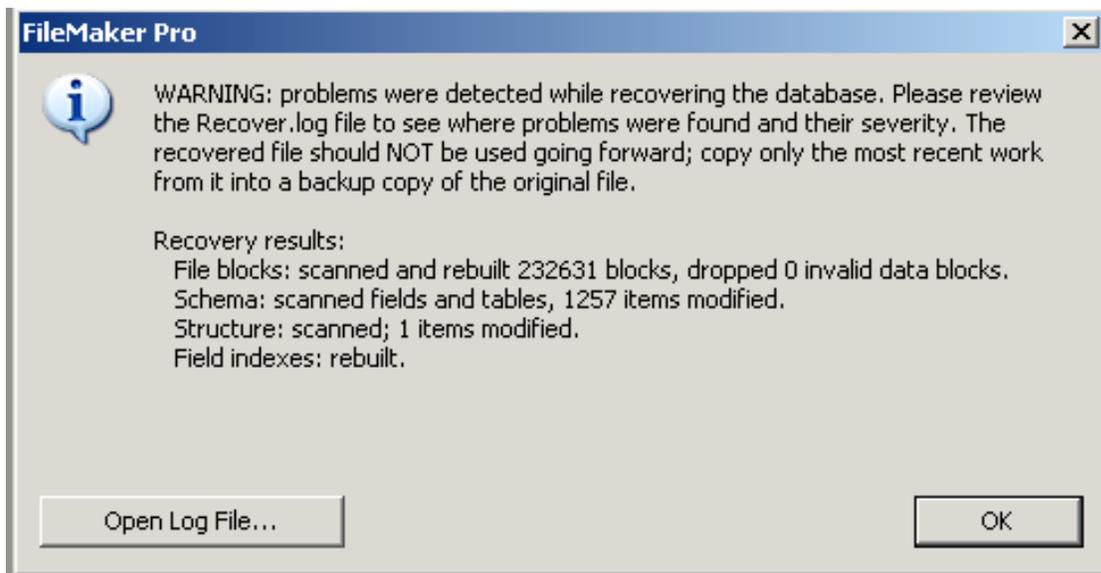


Figure 10: Potential problems remain after Recovery completes even though no data blocks were dropped. Note that the file should not be used.

So, what is the impact of all of this, and what does it mean? The most optimistic of these reports, the second one as shown in Figure 9, means that FileMaker Pro could not **detect** any further damage in the recovered file, **but FileMaker Pro cannot incontestably state** that there is no further damage especially if container objects are involved.

And that brings us back to *The Little Drummer Girl*. The safest course of action is to prevent damage to the file in the first place by following a rigorous regimen to protect it. Should damage occur and aberrant behavior start, run the Consistency Check, all the while bearing in mind its limitations. If a file is recovered and if FileMaker Pro reports a successful rebuild, then carefully consider the wording of that report's message:

...built a new database without detecting any problems. The new database is safe to use, though you should monitor the results carefully and make sure to keep up-to-date backups...

Now, this is the *most optimistic report* that the Recovery process will return. As noted earlier, it means that FileMaker Pro *detected* no further damage. But as also noted before, it is not an incontestable guarantee to produce or create a useable file free of every issue. So:

Yes, Repeat No.

And in such instances, developers and administrators must weigh the risks and benefits of continuing to use the new file as distinguished from a known good backup or known good clone of the structure.

Figure 10 tells a somewhat different story. The Recovery process you selected worked as best it could to retrieve elements of the file, and then reported:



...problems were detected while recovering the database....The recovered file should NOT be used going forward; copy only the most recent work from it into a backup copy of the original file.

As a result, developers can get back some otherwise lost work, but will need to decide *very carefully* where to place that retrieved work, e.g. several scripts written before a crash. Therefore:

Yes, Repeat No.

Remember again please that it is always possible that all Recovery options will fail, presenting this unhappy and unwanted dialog report:

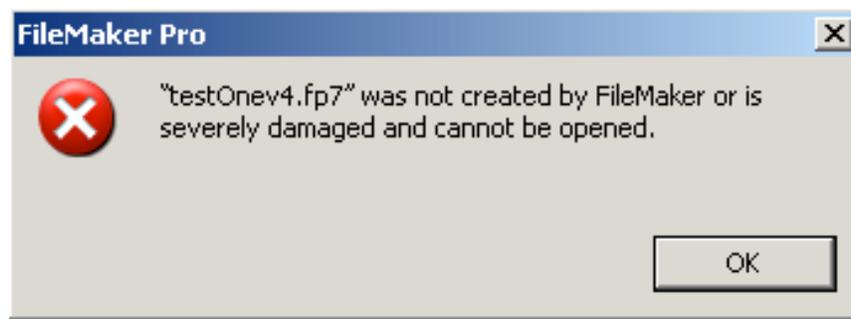


Figure 11: The file is too damaged for Recovery to work

That is precisely why, early on in this White Paper, I made these comments:

A rigorous backup regimen is an indispensable pre-condition for being able to utilize various components of the FileMaker Pro Recovery process as we will see in later sections of this White Paper. If you do not have good backups, your ability to fine tune the Recovery process will be seriously constrained and limited.

Effective and professional management of FileMaker Pro databases requires, not merely suggests, that rigorous and stringent steps be taken to safeguard the physical integrity of files and to avoid damage to them that would compromise their Integrity or Availability.

Summary and Review

In this White Paper, I have advanced several key propositions.

1. The best way for developers and administrators to manage FileMaker Pro File Recovery is to take rigorous and stringent steps to avoid file damage in the first instance. They do this by following established Best Practices for FileMaker Pro file *development* and subsequent *deployment* by FileMaker Server.
2. An effective, rigorous, and verifiable backup program is an indispensable ingredient in file maintenance and protection. Such a



- program extends the usability of the various granular options for file Recovery.
3. Developers and administrators have more control in FileMaker Pro 10 over the various Recovery features. They concomitantly have a greater responsibility to understand what those features do and to weigh the appropriateness of various options for using the resulting product of the Recovery regimen they select.
 4. Developers and administrators must weigh the adverse impact on their operations, people, organizations, and reputations that file damage will cause against the level of stringency associated with all available steps that are required to prevent that damage.
 5. Avoiding damage in the first place is the **best way** to manage the FileMaker Pro file Recovery process.

About The Author

Steven H. Blackwell is a **Platinum Member** of the FileMaker Business Alliance. From December of 1997 to April of 2007, he was a Partner level member of both the Claris Solutions Alliance (CSA) and the FileMaker Solutions Alliance (FSA), a world wide professional organization of developers of FileMaker Pro database products and solutions. He was among the very first class of thirty persons in the world designated as a FileMaker 7 Certified Developer™ by FileMaker, Inc. He also holds the FileMaker 8 Certified Developer™ designation. He also was among the first class of persons worldwide to receive the FileMaker 9 Certified Developer™ designation in February 2008. He is also a FileMaker Authorized Trainer™.

He is the author of the definitive volume *FileMaker Security: The Book*, available at www.filemakersecurity.com. He is also the author of, and instructor on, an entire set of videos on FileMaker Security released in June of 2007. [<http://www.vtc.com/products/Filemaker-Security-tutorials.htm>] He is also the author and instructor of the FileMaker Server 9 videos. [<http://www.vtc.com/products/FileMaker-Server-9-tutorials.htm>]

He is the co-author of the FileMaker, Inc. Technical Brief, *External Server Authentication* [<http://www.filemaker.com/support/upgrade/techbriefs.html>]. He is the author of the Technical Brief *Upgrading to FileMaker 7: How to employ the new, advanced Security System*, originally released to mark the advent of FileMaker® Pro 7 in March 2004 by FileMaker, Inc. and updated in mid-2008 for FileMaker® Pro 9. He is also one of the principal contributors to yet another 200 page technical document also released by FileMaker, Inc. in April 2004: *Upgrading to FileMaker 7: Migration Foundations and Methodologies* [<http://www.filemaker.com/support/upgrade/techbriefs.html>].

He is the Moderator of the on-line FM Forums Security and Server sections. FM Forums reaches over 40,000 FileMaker Pro developers and power users across the world.

Added information for and by Steven H. Blackwell can be found at:
<http://www.FMP-Power.com>
<http://www.filemakersecurity.com>



Acknowledgements

The generous and hard work over a number of years of a significant number of people helped me research and prepare this white paper. Without their assistance and interest, this resource would never have become available to the FileMaker community. All of us owe them a huge debt of thanks.

Rick Kalman, Senior Product Manager, Desktop Products, of FileMaker, Inc. was indispensable in the development of this White Paper.

Jon Thatcher, Senior Server Engineering Manager of FileMaker, Inc. was a major technical resource in guiding me through the complexities of the Recovery process and through the structure of FileMaker Pro files.

Wim Decorte, a Principal of FileMaker Business Alliance Platinum Member New Millennium Communications, Inc., provided invaluable insight about optimal best practices in an IT environment, especially related to FileMaker Server.

Barbara R. Levine, of FileMaker Business Alliance Platinum Member MicroServ, LLC, reviewed the manuscript with her usual sharp eye and offered many valuable suggestions to improve clarity and felicity of wording.

Daniel J. Mack, a Principal of FileMaker Business Alliance Platinum Member New Millennium Communications, Inc., has provided for a number of years the keenest insight and judgment on the critical importance of data integrity and business process integrity in FileMaker Pro database systems.

I would like to also acknowledge the work of the FileMaker, Inc. **System Engineering Team for North America**, both current and past, and their invaluable assistance in helping the developer community understand the behavior and complexities of the products. Andrew P. LeCates, their Manager, and former and current team members Tony Miller, Alexei Folger, Rosemary Tietge, Matthew O'Dell, Bill Heizer, and Rick Colcock all deserve the thanks and appreciation of the developer community.

