Technology Brief

Upgrading to FileMaker 7:

How to employ the new, advanced security system

About This Technical Brief

It is the intent of this technical brief to help the experienced FileMaker developer better understand the new, advanced FileMaker 7 security model. Reading this document will assist you in assessing the key features and benefits of the new security model and to plan, prepare for, and implement your strategy for migrating to FileMaker Pro 7. Authored by Steven Blackwell, FileMaker Solutions Alliance Partner and President and CEO of Management Counseling Services, this paper is part of a series of technical briefs written by developers for developers, to assist them in migrating 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.

Introduction

We live in a digital world, one with the Internet both as its principal highway and increasingly as a mirror reflecting society at large. The Internet and the services it offers are critical to businesses of every type and structure. Whether it is a university providing access to databases of scientific research data, a package delivery service providing tracking of shipments, a national trade association providing an on–line database of its members, a small, two–person business tracking accounts receivable and payable, or a manufacturing concern reporting to customers on order status—all these rely on rapidly accessible, constantly updated sources of information.

We want and we expect to be able to do anything on–line that we can do in the real world. We expect to be able to perform these activities with some level, expectation, and degree of certainty.[1] Yet reality is far different; the Internet is becoming increasingly less secure and more fraught with security threats. The limits of security are the limits of the Internet effectively speaking. And these limitations apply to FileMaker Pro® systems and to their developers and users.

Developers of software products should have every reason to expect that their proprietary intellectual property will remain safe and secure. Companies should have every expectation that their proprietary data will remain secure and safe from unauthorized disclosure. And companies also should have every expectation that only authorized individuals can add, modify, or delete information in their systems. Security then is important. But what are we trying to protect? What is vulnerable? Generally speaking, database security should address three specific issues:

- Protection of Intellectual Property
- Data Confidentiality
- Data Integrity

More broadly speaking, security is also part of a business continuity plan that assures the continued ability of a data–dependent enterprise to continue operations in the face of multiple failure points. Those issues are beyond the scope of this Technical Brief; however they are critically important.

This paper addresses five principal items of importance to FileMaker Pro solution developers and to IS/IT/DBA managers who work with FileMaker Pro and FileMaker Server:

- A brief review of the FileMaker Pro 6 security model and some of its concerns;
- An overview of the core features of new FileMaker Pro 7 and FileMaker Server 7 security system;
- A description of several significant security management issues the new system addresses;
- Several significant architectural and development technique issues for files that developers convert from earlier versions to FileMaker Pro 7; and,
- How the new system affects both the work and the business activities of three core constituent FileMaker Pro groups: *commercial solution developers, consulting developers, and IS/IT/DBA managers, particularly in enterprise workgroups.*

Table of Contents

Yesterday's Story

Previous versions of the FileMaker family of products were client–centric and relied on a trusted client model for authentication and password acceptance[2]. This led to some significant problems that in some instances required extensive workarounds to address. And more often than not, these workarounds lessened security rather than enhanced it. Network traffic was not encrypted. All this has now changed. And that change is pervasive and dramatic and powerful. *And as developers we have an unique opportunity to take advantage of the benefits that change offers.*

FileMaker Pro 7: A New Approach

The new FileMaker Pro 7 security system assumes that developers and IS/IT/DBA managers are serious about security. It does little good to have a new system if we do not use its features, and use them wisely and fully. ***The core purpose of the new system is to enforce the rules and processes necessary—on a case by case basis—to assure protection of intellectual property and to enforce data confidentiality and data integrity.***

The new FileMaker Pro 7 security system is its own layer of the database architecture; it is no longer part of the database schema layer: the layer where developers define such objects as tables, fields, file references, and relationships. And the implications of that separation are significant. Developers can grant classes of users {here called *Superusers*} the capability to create, delete, enable, disable, and reset accounts and passwords even while the files are opened and hosted for access by FileMaker Server 7 or FileMaker Server 7 Advanced. Changes take effect immediately; and they are propagated throughout the system. Moreover, these *Superusers* do **not** have to have access rights to the database schema to manage security, thus offering a significant level of protection to intellectual property, particularly that of commercial solution developers.

Accounts and Passwords: Credentials

The new FileMaker Pro 7 security system is account–based. The system relies on the authentication of user *credentials* to allow users access to the database at a prescribed privilege level defined by developer specified *Privilege Set*. Credentials consist of two elements: an account name and an account password, or in the case of external authentication, a Group name[3]. When a user's credentials are properly authenticated and the user is determined to be legitimate and valid, he or she can connect to the system at a specified level of access privileges defined by the Privilege Set. Figure 1 illustrates this process.

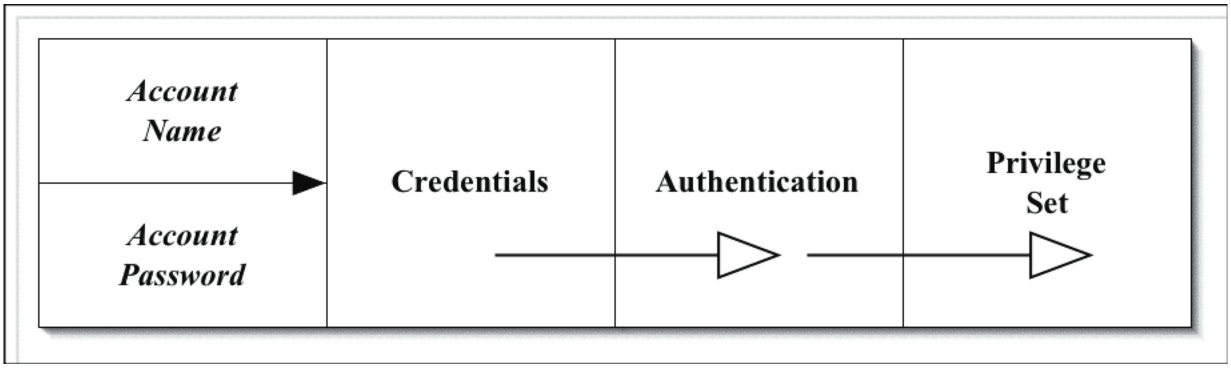


Figure 1. The core concept of the security system.

There are important rules for the construction of credentials. *Account names must be unique; however they are not case–sensitive*. When establishing an account in multiple files, developers should take care however to make the account names exactly the same. This will prevent confusion, and it provides more order to the system. *Passwords are case–sensitive; however, they need not be unique*. At first blush, this may seem likely to cause problems; however that is not the case, as I will shortly explain. Note that this system is a complete change from the model in prior versions of the product. Developers can specify that passwords be of a minimum length, and that passwords expire after a fixed interval of time. If a user forgets his or her password, and that will happen, an administrator with proper privileges can recreate the account, reset the password, etc. and then require the user to select a new one.

Developers define accounts and passwords from the *File* menu; developers can also grant account management privileges to *Superusers*, and I will explain that later in this paper. From the *File* menu select [File-Define-Accounts & Privileges] to reveal a tabbed interface similar to this one:

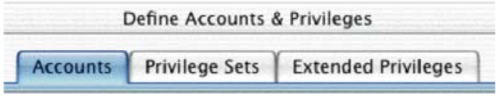
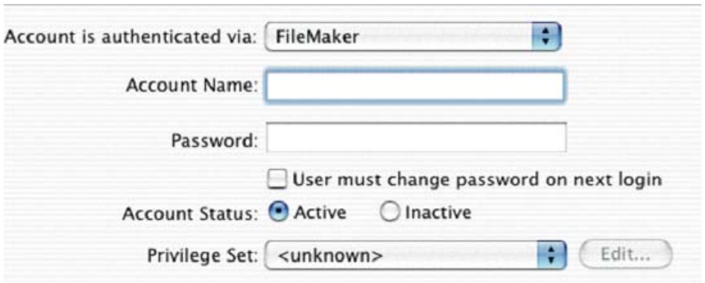


Figure 2. Define Accounts & Privileges Tab

Selecting the Accounts tab produces a window where the account definition and authentication options are set:



Account is authenticated via: FileMaker

Account Name:

Password:

□ User must change password on next login

Account Status: ● Active ○ Inactive

Privilege Set: <unknown>

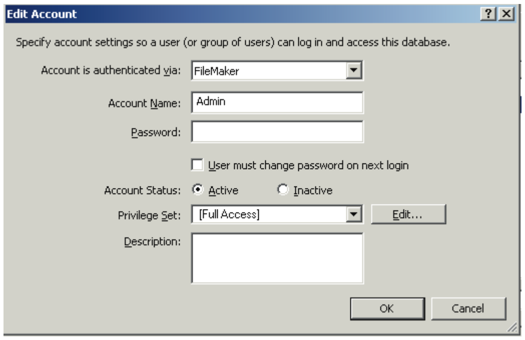
Figure 3. The Edit Account Window.

Here the developer gives the account a name and a password. The developer can mandate that the user change the password at the next logon by checking the checkbox.

Generally speaking, security is enhanced when the user is required to assume responsibility for the ownership of the password. While administrators can reset the password when the user forgets it—and that may happen—the system is made more secure when no one but the individual user knows his or her password. Likewise some care needs to be taken in defining account names. Many organizations utilize a standard nomenclature for account names. For example: *SmithA* or Smith_a could be standard account names for a user Andrew Smith. Thus, an account name might be easy to guess, providing a security vulnerability. User selected passwords lessen that vulnerability as do variants of an account name nomenclature such as *SmithA#$5*. Strong passwords are eight or more characters in length and mix alphanumeric and non–alphanumeric

characters[4]. They are easy to remember, but hard to guess. FileMaker Pro 7 supports passwords or passphrases of up to 100 characters in length, including spaces. An example of a passphrase would be an *adaptation* of an easy to remember, but hard to guess, line of poetry or famous quotation, such as:
Able was I, ere I saw Elba; but Napoleon lost at Waterloo

When a developer first creates a new FileMaker Pro 7 file, the application creates a default account called *Admin* without any password and assigns it to the [Full Access] Privilege Set as shown in Figure 4. FileMaker Pro 7 also sets an automatic log–on[5] with the Account Name *"Admin"* and a blank password.This enables the developer to work on the file. I recommend that as a first step that the developer rename this default account to something other than *Admin* and assign it a password. Otherwise, it remains at default and is therefore highly susceptible to being guessed, thus granting access to the file. Developers should safeguard the credentials linked to [Full Access] Privilege Sets. If the password is lost or forgotten, it cannot be retrieved, even by FileMaker Inc.



Edit Account

Specify account settings so a user (or group of users) can log in and access this database.

Account is authenticated via: FileMaker

Account Name: Admin

Password:

□ User must change password on next login

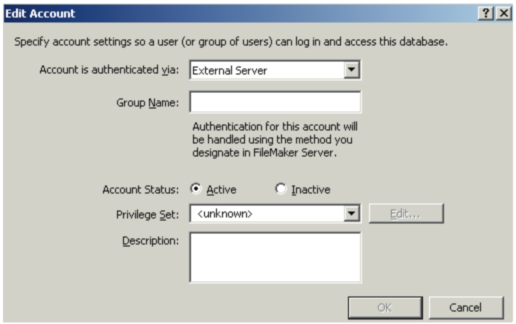
Account Status: ● Active ○ Inactive

Privilege Set: [Full Access]

Description:

Figure 4.The default Admin account with [Full Access] privileges.

In the Edit Account window the developer also selects the authentication method: *FileMaker or External Server.* If the developer selects the latter option, the screen changes as shown in Figure 5.



Edit Account

Specify account settings so a user (or group of users) can log in and access this database.

Account is authenticated via: External Server

Group Name:

Authentication for this account will be handled using the method you designate in FileMaker Server.

Account Status: ● Active ○ Inactive

Privilege Set: <unknown>

Description:

Figure 5. Edit Accounts with External Authentication Options.

Note that the Account Name has changed to Group Name. Here you would enter the name of a Group from the external domain,[6] and now FileMaker Server 7 manages the authentication externally. There are several important items to note about these account definition and authentication options:

1. All files must have at least one internally authenticated {by FileMaker Pro} [Full Access] account.[7]
2. I would recommend never authenticating an account with [Full Access] privileges by external methods. If a physical copy of the file were to be obtained, the Domain system could be recreated, and the Domain account spoofed, granting unrestricted access to the files.
3. Good security practice requires that account names are intended for *individual users*. **Users should not share or divulge their credential information.** Individuality is at the heart of a security system that enforces the three pillars of security: intellectual property protection, data confidentiality, and data integrity.

When is it appropriate to use one method of authentication as distinguished from another? In some instances, developers will employ the internal FileMaker authentication because the solution will be deployed in an organization without a Domain structure. However in many others instances both consulting developers and internal, in–house developers will likely adopt the external authentication method in order to leverage existing IS/IT assets and to standardize multiple account and multiple solution management. In the external authentication scenario, the user's Domain credentials are employed for purposes of granting access to the FileMaker Pro database at a specified level of access. Note that a file can have both internally authenticated accounts and externally authenticated accounts as well. In FileMaker Server 7, the server administrator can select which option to use: *FileMaker accounts only or FileMaker and External Server accounts*. I will discuss this further later in this paper.

The external authentication option does mean that FileMaker Pro 7 in conjunction with FileMaker Server 7 will support single–source log–ons, sometimes called universal authentication log–on or single sign–on. This is a commonly employed technique in IS/IT system and network management. The general belief is that it simplifies user credential management activity by requiring the user to remember only one set of credentials to access digital assets and network based assets. While this belief is almost certainly a correct one, nevertheless it does transfer the security of the database to something outside of FileMaker Pro. Developers may wish therefore to learn more about network security and authentication generally.[8]

Privilege Sets

Once a developer has created a new FileMaker Pro 7 file, he or she can assign any developer created accounts to one of the default Privilege Sets. There are two types of Privilege Sets: [Full Access] and all others that are *subordinate*. These subordinate Privilege Sets, whether default or developer created, have some sort of restrictions attached to them. A developer **cannot** create a [Full Access] Privilege Set; only the default one is available. There are two default Privilege Sets other than the [Full Access]: [Data Entry Only] and [Read–Only Access]. Both are subordinate. Developers should carefully examine the elements of both of these default subordinate Privilege Sets before assigning them to accounts. In many instances, despite their names, they may contain a different level of privilege than the developer desires for accounts. Therefore, the developer may elect to use a custom designed account for "Data Entry" and for "Read Only" accounts.

Developers can and will create customized subordinate Privilege Sets, for it is by creating these that the full power and flexibility of the new security system is brought to bear. Privilege Sets are at the heart of the security enforcement schema of FileMaker Pro 7. They determine what actions and rights a user has within the respective file and to all the tables contained in that file.

In the *Define Accounts & Privileges* tab {Figure 6}, selecting the second tab Privilege Sets opens a window similar to Figure 7 where the developer creates a customized subordinate Privilege Set.

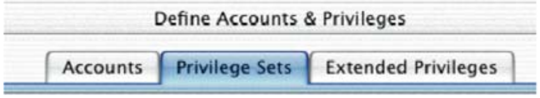
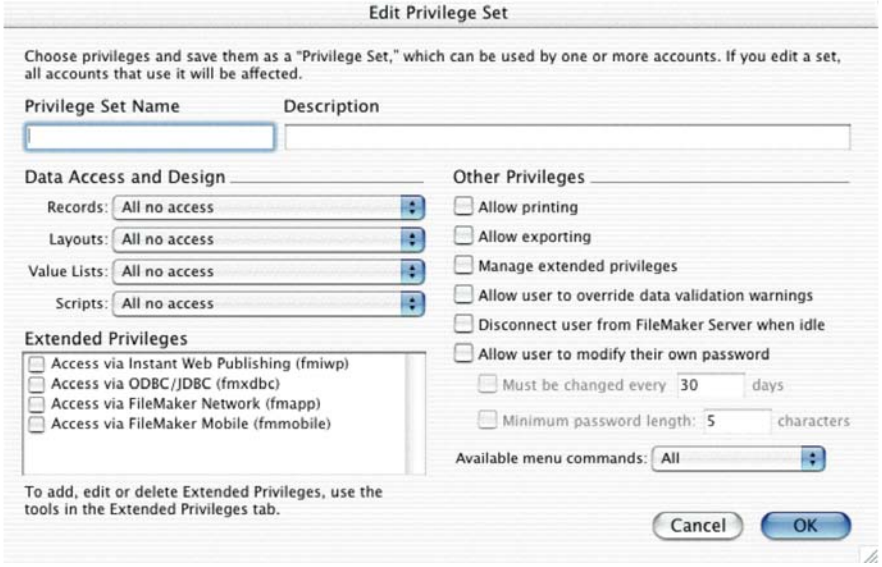


Figure 6. Define Accounts & Privileges Tab



Edit Privilege Set

Choose privileges and save them as a "Privilege Set," which can be used by one or more accounts. If you edit a set, all accounts that use it will be affected.

Privilege Set Name

Description

Data Access and Design

Records: All no access

Layouts: All no access

Value Lists: All no access

Scripts: All no access

Extended Privileges

□ Access via Instant Web Publishing (fmiwp)
□ Access via ODBC/JDBC (fmxdbc)
□ Access via FileMaker Network (fmapp)
□ Access via FileMaker Mobile (fmmobile)

To add, edit or delete Extended Privileges, use the tools in the Extended Privileges tab.

Other Privileges

□ Allow printing
□ Allow exporting
□ Manage extended privileges
□ Allow user to override data validation warnings
□ Disconnect user from FileMaker Server when idle
□ Allow user to modify their own password
□ Must be changed every 30 days
□ Minimum password length: 5 characters

Available menu commands: All

Figure 7. Edit Privilege Set.

The first thing to note about this Privilege Set window is that most options are *closed* by default. This is a significant change from earlier versions where the solution was open, and the developer had to close access object by object. FileMaker Pro 7 employs a default of closed access, and developers must specifically grant access to a wide variety of objects and functions.

Each account has *one, and only one*, Privilege Set attached to it. A given Privilege Set can have more than one Account attached to it however. This too is a distinct change from prior versions, where passwords could be assigned to different Groups. Note the various areas of the Privilege Set window: *Data Access and Design,* *Extended Privileges, and Other Privileges*. In the subsequent discussion on granularity of access, I will cover these in more detail. Also note the text box labeled Description, where comments or a description of the purpose of the Privilege Set can be entered. This new feature is very helpful for architecture management and for technical documentation of solutions.

Extended Privileges

Figure 8. Extended Privileges tab.

Figure 8 shows the last tab in the *Define Accounts & Privileges* options. Clicking on this area reveals a window where users can be authorized to assign Extended Privileges to various Privilege Sets and by extension to the accounts that are attached to that Privilege Set. Most of these deal with network connectivity options such as access from FileMaker Server, Instant Web Publishing, Custom Web Publishing using the new Web Publishing Engine in FileMaker Server 7 Advanced, ODBC/JDBC connections to the database, and FileMaker Mobile connections. Additionally, developers can define custom Extended Privileges for use with various external or internal modules.

Planning and Order Of Security Schema

The new FileMaker Pro 7 security schema requires developers to plan for security in a different fashion than they did in the past. This is a nuanced issue, and it is one that requires more "boots–on–the–ground" experience before we can recommend any comprehensive set of best practices.

As a general rule, it may prove easier, however, once solution specifications are established, to create several custom defined subordinate Privilege Sets and attach to each an easily recognizable Account Name and password for testing during development[9]. That allows for tweaking of the Privilege Set and for proper assignment of privileges to objects at a fine granular level if desired.

Complex solutions—and even those of considerably less complexity—will benefit however from having a section in their design specifications that explicitly addresses access privileges. Defining those Privilege Sets can be a challenge. How can developers best plan for effective use of the new security system? There are several well recognized and distinct processes for access management:

- Mandatory Access Control;
- Discretionary Access Control (now employed extensively in FileMaker Pro 6 files);
- Rule Based Access Control; and,
- Role Based Access Control.

In the Role Based approach, developers would construct a custom designed subordinate Privilege Set for each identified role in the database, and then assign as many accounts to that Privilege Set as needed to accommodate users fulfilling that role.[10]

New Functions Related To Security Management

FileMaker Pro 7 has a number of new functions related to security management. These return information about the Account used to access the file. These are Get Functions, replacing the Status Functions. These include Get (AccountName)[11] Get (PrivilegeSetName), and Get (ExtendedPrivileges). Get (UserName) survives from prior versions, although its use and usefulness is diminished[12]. Additionally there are some CPU and network based functions that have security use, including Get (SystemNICAddress) and Get (SystemIPAddress). The results returned from these functions that can be used are scripts, calculations, Record Level Access tests, and similar areas to help identify the account and create conditions based on that identification.

Granularity of Access

Granularity refers to the level of discrete, differentiated access control that the security system grants to a wide variety of FileMaker Pro objects and functions:

Objects

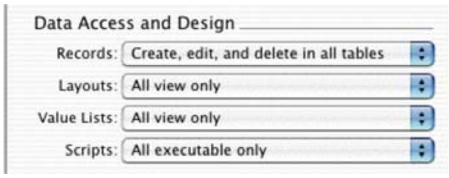
Table
ScriptMaker™ script creation
ScriptMaker scripts access
Value list creation
Value list access
Layout creation
Layout access
Record
Field

Functions

Print
Export
Manage Own Password
Database sharing options including networking, ODBC/JDBC, Instant Web Publishing, Custom Web Publishing, and FileMaker Mobile
Idle activity disconnect
Schema access under controlled conditions
Account management
External API manipulation
Extended Privilege management.

Extended Privilege *management*, as noted in a prior section, should be distinguished from the Extended Privilege itself, per se. The management refers to the ability to enable or disable the Extended Privilege as well as the ability to create new custom privileges, to delete them, and to assign and unassign Extended Privileges to specific Privilege Sets. I will explain more about Extended Privileges later in this Technical Brief.

For developers who do not need or want to customize settings at a high level of granularity, FileMaker Pro 7, as did its predecessor versions, offers a variety of standard settings for security that enable rapid assignment of accessibility options to custom create subordinate Privilege Sets. Figure 9 illustrates one example of this.



Data Access and Design

Records: Create, edit, and delete in all tables

Layouts: All view only

Value Lists: All view only

Scripts: All executable only

Figure 9. Standard Access options.

Granularity is a very important concept, and its extensive use greatly empowers the developer to control access to objects and functions in the file, thus enforcing the Three Pillars of Security: intellectual property protection, data confidentiality, and data integrity. For these objects and functions there are at least **four levels of control** that can be applied object by object or to an entire category. This is done Privilege Set by Privilege Set for any custom created subordinate Privilege Set, further enhancing granularity of access control. The levels:

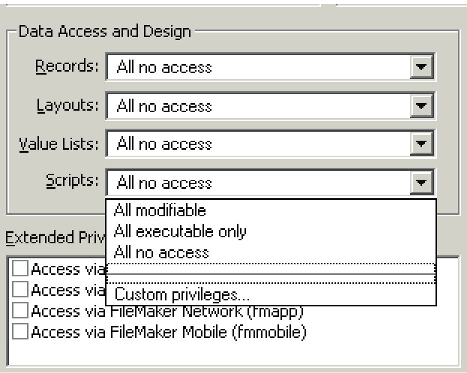
- *Create*, except for tables, fields, and relationships, although these can be created under controlled circumstances. [13]
- *Modify,* including delete.
- *Read only,* or in the event of scripts, execute only.
- *No Access.*

These controls can be applied selectively to developer created objects as distinguished from ones that end users or administrators might subsequently create. **This means that developers can assign an administrator, or even an end user, the ability to create new layouts, scripts, value lists, and scripts without being able to modify existing ones.**

There are some rather significant implications flowing from the developer's ability to allow an administrator to create new scripts without being able to alter ones the developer created. Such granularity of access protects developer proprietary intellectual property, particularly for commercial solution developers. It also protects proprietary business processes from disruption while allowing administrators to customize some additional functionality.

Scripts have special functionality as befits their core role in the FileMaker Pro process. Developers can for each and every subordinate custom designed Privilege Set designate a specific script to be *Modifiable*, meaning that the user can change it. Or the script may be marked as *Executable Only*, meaning that the user has access to it but cannot see its logic and individual steps and can not alter it. Or, the developer can mark the script as *No Access*, meaning that the user neither sees it nor knows it even exists for that matter. It does not appear in the list of scripts even when ScriptMaker™ is opened. And the same user can create new scripts, but not even see the ones marked *No Access* for a user specific Account's attached Privilege Set.

Figure 10 shows the options for scripts along with the *Custom Privileges* option. This allows for script by script differentiation of access options. FileMaker Pro 7 has similar levels of granularity for most other objects, as noted in the table.



Data Access and Design

Records: All no access

Layouts: All no access

Value Lists: All no access

Scripts: All no access

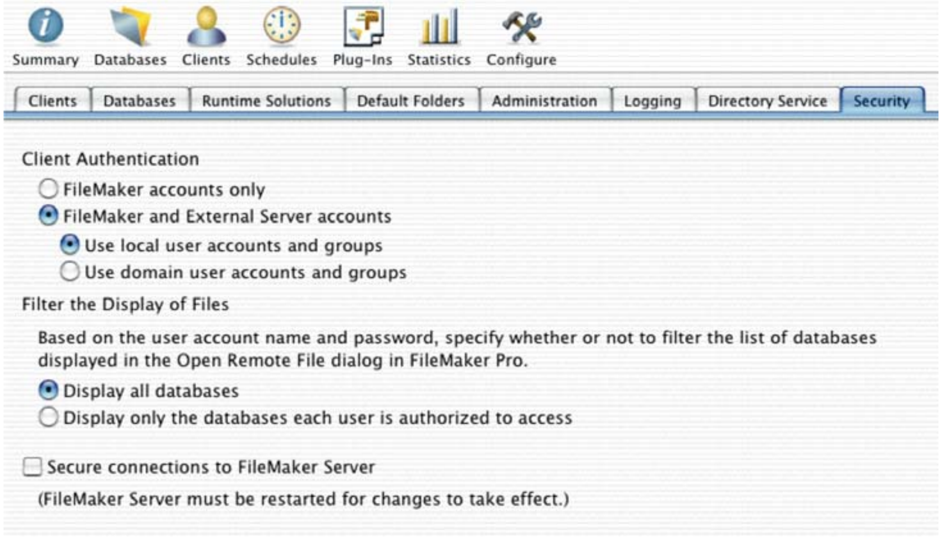
Figure 10. Highly granular Script accessibility options.

FileMaker Server 7

FileMaker Server 7 and FileMaker Server 7 Advanced play important roles in the new security system in three critical areas: authentication, file visibility, and data encryption.

Both the Macintosh OS X version and the Windows 2000 Server/Windows 2003 Server versions of FileMaker Server 7 have built–in security controlling authentication requirements for accessing the daemon or service. The management of such items is described more fully in a separate Technical Brief on FileMaker Server 7, but basically both can be made to require log–on authentication for their access and administration. Maintaining the physical security of the CPU's running FileMaker Server 7 and the physical security of the files being hosted for access is very important. Such precautions as turning off OS level file–sharing, keeping the CPU's in a locked and secured environment, and properly securing and accounting for the location of all back–up copies of the files can contribute to enhanced security. Additionally, FileMaker Server 7 has extensive logging capabilities[14] both for itself and for its hosted files; this assists in the critical functions of process monitoring and access monitoring.

Figures 11A and 11B illustrate the Macintosh OS X and Windows 2000 Server/2003 Server security panels for FileMaker Server 7.



Client Authentication

○ FileMaker accounts only

● FileMaker and External Server accounts

● Use local user accounts and groups

○ Use domain user accounts and groups

Filter the Display of Files

Based on the user account name and password, specify whether or not to filter the list of databases displayed in the Open Remote File dialog in FileMaker Pro.

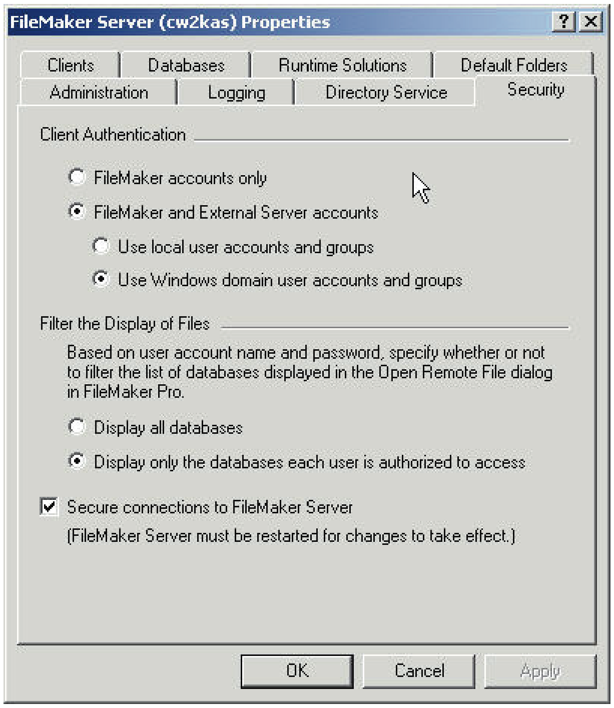
● Display all databases

○ Display only the databases each user is authorized to access

□ Secure connections to FileMaker Server

(FileMaker Server must be restarted for changes to take effect.)

Figure 11A. Macintosh OS X FileMaker Server security management tab.



Client Authentication

○ FileMaker accounts only

● FileMaker and External Server accounts

○ Use local user accounts and groups

● Use Windows domain user accounts and groups

Filter the Display of Files

Based on user account name and password, specify whether or not to filter the list of databases displayed in the Open Remote File dialog in FileMaker Pro.

○ Display all databases

● Display only the databases each user is authorized to access

☑ Secure connections to FileMaker Server

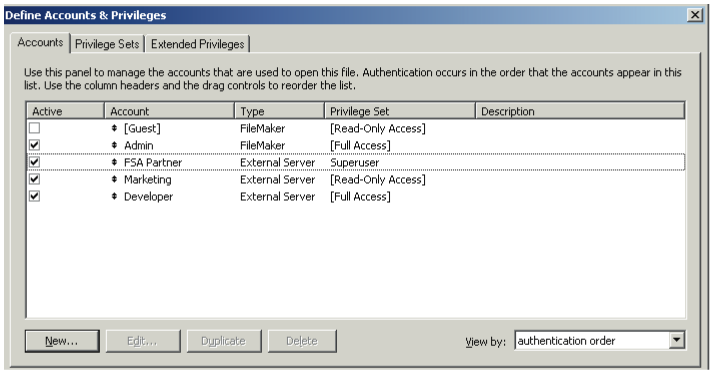
(FileMaker Server must be restarted for changes to take effect.)

Figure 11B. Windows 2003 Server FileMaker Server security management tab.

As we saw earlier, when a FileMaker Pro 7 file has an account designated to be authenticated externally, FileMaker Server 7 performs this task. Note the option *Client Authentication*. On either version the server administrator can select FileMaker accounts only or *FileMaker and External Server accounts.* Additionally, if selecting *External Server Accounts* the administrator can select from *local user accounts and groups* or from *domain user accounts and groups*. The distinctions between local and global domains are more fully explored in the FileMaker Server 7 Technical Brief, but basically this option enables an administrator to remove the FileMaker Server 7 CPU from the enterprise domain, but set up user accounts and groups on that CPU for use in external authentication.[15]

There are several architectural and deployment considerations developers and IS/IT/DBA managers should bear in mind when utilizing external authentication. First, selecting the option *FileMaker accounts only* effectively disables externally authenticated accounts in a particular FileMaker Pro 7 database file.

Second, in enterprise Domains, users typically belong to multiple Domain Groups[16]. This raises the question of which Group the user is to be authenticated against when accessing the database files. The user *Jane Smith* may be a member of the Marketing Group, the Developer Group, and the FSA Partners Group. Each of these groups has an account in the database file, and each account has markedly different privileges assigned through its attached Privilege Set. How is access determined? The answer is that the developer selects the authentication order in the *Accounts* tab of the *Define Accounts & Privileges* section of the database. The **first** matching account found in the authentication order is the one used to determine privileges. Figure 12 illustrates this concept, showing that *Jane Smith* will connect with account *FSA Partners* with the Privilege Set *Superuser* if externally authenticated. Developers, in conjunction with IS/IT/DBA managers must take steps to assure that expected levels of access occur when faced with the possibility of multi–Group membership. Such users could, of course, access the files with an internally authenticated account if need be.



Use this panel to manage the accounts that are used to open this file. Authentication occurs in the order that the accounts appear in this list. Use the column headers and the drag controls to reorder the list.

Active	Account	Type	Privilege Set	Description
□	[Guest]	FileMaker	[Read-Only Access]	
☑	Admin	FileMaker	[Full Access]	
☑	FSA Partner	External Server	Superuser	
☑	Marketing	External Server	[Read-Only Access]	
☑	Developer	External Server	[Full Access]	

View by: authentication order

Figure 12. Authentication Order determines account and corresponding Privilege Set selected from external authentication when a user belongs to multiple Domain groups.

Refer again to Figures 11A and 11B for the second option, *Secure Connections to FileMaker Server.* This is a binary option; it is either on or off. When enabled, this option encrypts the data traffic between FileMaker Pro 7 clients and FileMaker Server 7. It also encrypts data traffic between FileMaker Server 7 and the new Web Publishing Engine for both Instant Web Publishing and Custom Web Publishing. I will have some additional information about this option's effect on Web-based account access in the following section on the Unified Model.

The third FileMaker Server 7 based option shown in Figures 11A and 11B relates to the display of file names shown in the *Open Remote* menu option [File-Open Remote…] in FileMaker Pro 7. At its simplest level this option causes all databases whose names are supposed to be displayed to appear if the *Display All Databases* option is selected. Otherwise, if the administrator selects the option *Display only the databases each user is authorized to access*, the user must first have an authenticated account in the database in order to access it in the first place.

When this filtering or visibility option is in force and a user attempts a connection to the server, FileMaker Pro 7 will attempt to utilize saved account information for the user seeking access. On Macintosh OS X the Keychain Manager is used; on Windows 2000 Professional and Windows XP Professional, the user credentials are used. If there are no valid matches, the user receives a modal dialog window requesting that he or she enter credentials {an account name and account password} to view databases hosted by the server. This same option can be invoked by holding down the OPTION key on Macintosh OS X or the SHIFT key on Windows 2000 Professional or Windows XP Professional at the time the user selects the server name from the *Open Remote…* menu item. If a user enters incorrect information, he or she will encounter another modal dialog box asking them to relog into the server with different credentials.

Web Based Access: Unified Security Model

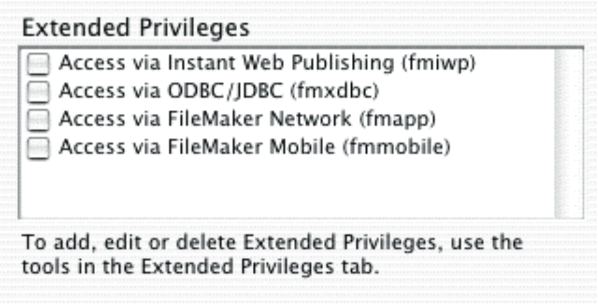
The FileMaker Pro 7 security features carry through in a unified model to users accessing the databases *via* web browsers. Developers can enable the ability of web-based users to create their own accounts seamlessly and pass them into the database. Such a feature has many uses, including, for example, on–line registration systems. Custom designed subordinate Privilege Sets can have web user accounts attached to them, just as any other accounts. The privileges enforced by a particular Privilege Set are carried through to web users for both Instant Web Publishing and Custom Web Publishing. Additionally if a LAN-based user has been given permission to access a file from the web *via* a browser, the user LAN-based FileMaker Pro client privileges carry through to that web-based access. This allows for the ability to impose a robust set of access rules account by account, table by table. Additionally, the filtering or visibility option carries through to web based access as well.

Administrators can also manage accounts in LAN–WAN based FileMaker Pro files from web interfaces and access if permissions have been properly structured to allow this feature. There may be business or policy decisions that must be addressed in this process; however, it can be done.

Web-based users can take advantage of the encryption capabilities of FileMaker Server 7. Toggling the encryption to "on" creates the encrypted channel between FileMaker Server 7 and FileMaker Pro 7 clients; it also creates an encrypted channel to the Web Publishing Engine. Developers and IS/IT/DBA managers should consult the FileMaker Server and FileMaker Web Publishing Technical Briefs for information about configuration of the Web Publishing Engine. Both Apache and Microsoft IIS support SSL connections from modern web browsers. This then leaves the connection between the Web Publishing Engine and Apache or IIS as the remaining part of the total data channel requiring protection. There are a variety of approaches available to protect this link.[17]

In order for FileMaker Server 7 and FileMaker Server 7 Advanced to allow for web–based connections, the server must be instructed to allow such connections and have the appropriate license key granting that privilege to do so installed. Refer to the Technical Briefs on FileMaker 7 Web Publishing and on FileMaker Server 7. An analogous situation exists for ODBC/JDBC connections where the ability to accept such connections must be specifically enabled in the *Clients* tab of FileMaker Server 7 and FileMaker Server 7 Advanced. Additionally, for Instant Web Publishing, Custom Web Publishing, and ODBC/JDBC connectivity, developers, or in some instances *Superusers*, must enable the appropriate Extended Privilege in the file to be shared. Such Extended Privileges can be configured on an individual Privilege Set basis.

Let's look again at a portion of the Edit privilege Set window we saw in Figure 7. Figure 13 shows that portion, the one devoted to the four default Extended Privileges.



Extended Privileges

□ Access via Instant Web Publishing (fmiwp)
□ Access via ODBC/JDBC (fmxdbc)
□ Access via FileMaker Network (fmapp)
□ Access via FileMaker Mobile (fmmobile)

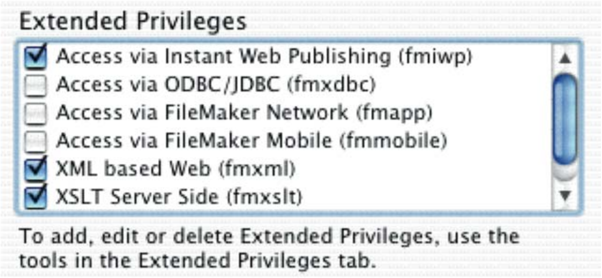
To add, edit or delete Extended Privileges, use the tools in the Extended Privileges tab.

Figure 13. Default Extended Privileges

Developers would check the options for Instant Web Publishing, keyword fmiwp, and ODBC/JDBC, keyword *fmxdbc*, for the Privilege Sets, including the [Full Access] one to which they wish to grant that access privilege. Then any accounts attached to that Privilege Set could access the files in that matter if FileMaker Server 7 is hosting the file. But—to be clear—there is an entire chain of security control that must be observed:

- Extended Privilege enabled for specific Privilege Set;
- Authenticated user credentials {account name and account password} for account attached to that Privilege Set; and,
- FileMaker Server properly licensed and configured to allow Instant Web Publishing, Custom Web Publishing, and ODBC/JDBC connections.

If any of these are not activated, e.g. the Extended Privilege, the file can not be accessed with an account attached to that Privilege Set even if the file is hosted by FileMaker Server. If a developer wants to grant access privileges to a file via Custom Web Publishing, the developer must create two custom Extended Privileges with the keywords *fmxml* and *fmxslt* depending on the type of Custom Web Publishing access desired.[18] Figure 14 shows these privileges enabled, along with the Instant Web Publishing one.



Extended Privileges

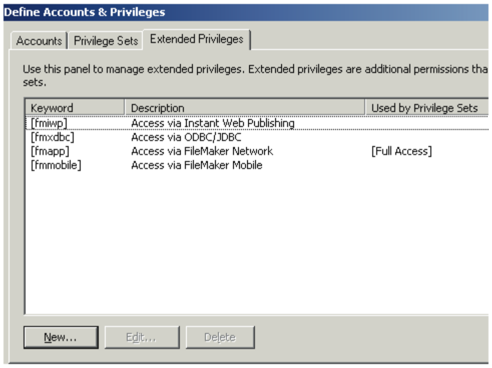
☑ Access via Instant Web Publishing (fmiwp)
☐ Access via ODBC/JDBC (fmxdbc)
☐ Access via FileMaker Network (fmapp)
☐ Access via FileMaker Mobile (fmmobile)
☑ XML based Web (fmxml)
☑ XSLT Server Side (fmxslt)

To add, edit or delete Extended Privileges, use the tools in the Extended Privileges tab.

Figure 14. Custom Web Publishing Extended Privileges along with Instant Web Publishing ones.

Figure 8 showed the Extended Privileges Tab. Here it is again for ready reference:

Clicking on that tab takes the developer to a window similar to the one in Figure 15. From within this window the developer or the *Superuser* with appropriate privileges can define new, custom Extended Privileges and assign them to various Privilege Sets. Thereby they will be available to accounts attached to those Privilege Sets. Figure 7 shows where to select the option to select the rights to manage Extended Privileges for a custom subordinate Privilege Set. Checking the *Other Privileges* option *Manage extended privileges* authorizes this action. Any user with an authenticated account attached to that Privilege Set could administer the Extended Privileges.



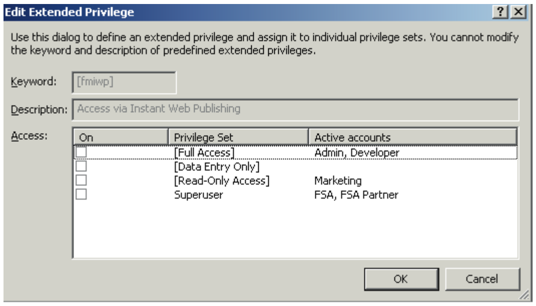
Use this panel to manage extended privileges. Extended privileges are additional permissions tha sets.

Keyword	Description	Used by Privilege Sets
[fmiwp]	Access via Instant Web Publishing	
[fmxdbc]	Access via ODBC/JDBC	
[fmapp]	Access via FileMaker Network	[Full Access]
[fmmobile]	Access via FileMaker Mobile	

Figure 15. Define Extended Privileges Window

When the developer or the *Superuser* clicks either the button marked *New…* or the one marked *Edit…* a dialog window similar to the one in Figure 16 appears. Developers can define new custom Extended Privileges and assign them to accounts in this window.   Developers should take note that granting a user access to this feature permits unassigning Extended Privileges from all accounts in the file, including those associated with

[Full Access] Privilege Sets. Obviously, in some instances this behavior may not be desirable, and so developers must carefully weigh the implications of granting rights to manage Extended Privileges. But in enterprise deployments it likely will be a necessity to grant these rights at least to administrators or to *Superusers*.



Edit Extended Privilege

Use this dialog to define an extended privilege and assign it to individual privilege sets. You cannot modify the keyword and description of predefined extended privileges.

Keyword: [fmiwp]

Description: Access via Instant Web Publishing

Access:

On	Privilege Set	Active accounts
□	[Full Access]	Admin, Developer
□	[Data Entry Only]	
□	[Read-Only Access]	Marketing
□	Superuser	FSA, FSA Partner

Figure 16. Editing and assigning Extended Privileges.

FileMaker Pro 7 Addresses Key Security Management Issues

The new FileMaker Pro 7 security system improves the features in earlier versions of the products. The principal ones are access management, network traffic interception, granularity of control over objects and functions, password extraction, and text editor manipulation of FileMaker Pro files.

Account Management

In prior versions of FileMaker Pro, it was more difficult for developers and administrators to manage multiple passwords and Groups, especially across multiple files in solutions. In the new FileMaker Pro 7, files can have multiple tables. At first blush this might seem to solve the multi–file security system management problems. Some will want merely to collapse a multi–file solution into a multi–table, *single–file* solution where all the security options can be centrally managed.While there are many instances where appropriate architecture may mandate a *single–file, multi–table* approach, there are likely to be just as many instances where architecture and business processes mandate *multi–file, multi–table* solutions. Moreover, converting a FileMaker Pro 6 multi–file solution to FileMaker Pro 7 results in a multi–file FileMaker Pro 7 solution. *Multi–file* security schema management is still a requirement.

Developers can now grant administrators or other *Superusers* the capability to manage accounts while the files are open. Changes made in this process take effect immediately. A *Superuser* can create a new account, delete an existing account, disable an existing account, enable an existing account, or reset an account password. Such a process can allow the *Superuser* to choose the account password or allow the *Superuser* to pass along the requirement that the user create a new password on next log–on.

This capacity can extend across all files in a solution. ***The Superuser does not have to be granted access to the database schema such as tables, fields, and relationships to manage accounts.*** Commercial solution developers can thus construct an account management feature in their solutions that helps protect the developer's of intellectual property.

Unlike the developer who defines account names and passwords from the *File* menu select [File-Define-Accounts & Privileges], the *Superuser* manages account names, account passwords, and account status through scripted activity. A new category of ScriptMaker script steps called *Accounts* enables this type of management control. Using the UI options provided by the *Show Custom Dialog* ScriptMaker step, the administrator or *Superuser* could pass the variables to the ScriptMaker script step. e.g. *Create New Account.* This works for one account at a time.

But if the *Superuser* wants to create fifty new accounts *simultaneously* in one or more files, the appropriate variables can be passed from file to file with the action occurring sequentially one file after another, one account after the other. While on its surface this sounds complex and time–consuming, it really is not. I created 1000 new unique accounts in a single file in less than two minutes using this automated approach. This scripted and automated process simply passed the required variables of *Account Name* and *Account Password* one at a time from a control file to the target file.

In addition to the account management ScriptMaker script steps there is also a new ScriptMaker step: *"Re-login"* that is extremely useful for security management. Using this step the *Superuser* can perform a relog–on to a file with a new, different account without having to close the file.

But how can a *Superuser* or an administrator without any access to the file at a [Full Access] Privilege Set level control security management? Developers can temporarily grant a category of users the ability to perform actions by a script they would not otherwise be able to perform. Each script has an option to **"Run script with full access privileges"** that the developer can toggle on a script by script basis. Restricting access to the account management scripts to just a custom designed subordinate Privilege Set used by the *Superuser*'s account but making those scripts run with the full access enables the account management to take place. Figure 17 illustrates the "Run script with full access privileges" option toggle at the bottom of the *Edit Script* window.

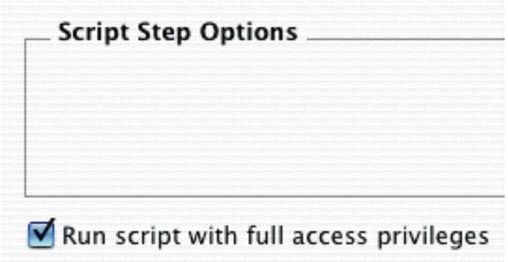


Figure 17. Script full access option.

When activated, the script runs[19] as if the user were connected to the file with an account having the [Full Access] Privilege Set enabled. It is important to note that this confers power onto the *script*, not onto the user. Combined with the ability Privilege Set by Privilege Set to make a script accessible, the developer can exert a very fine level of control over who can perform a particular function. Using this same function, a developer can also grant a *Superuser* connected with an account attached to a custom defined subordinate Privilege Set the capability to access the *Define Database* or *Define File References* functions. Obviously, this privilege should be granted only rarely and should be used with caution.

Encrypted Network Traffic

In previous versions of the products, packet sniffing software could detect encoded FileMaker Pro password packets moving across networks, particularly when the previous versions of FileMaker Server sent these in a encoded fashion to a guest for verification before granting access to a file.

In FileMaker Pro 7, however, authentication occurs at the server level, not at the client level. And the passwords are not stored in the files,[20] so interception is much harder. Additionally, as I mentioned in the section on FileMaker Server, network data traffic can now be sent in encrypted packets. FileMaker Pro 7 and FileMaker Server 7 use industry standard, widely tested and accepted security algorithms.

The new versions of FileMaker Pro and FileMaker Server use the industry standard TripleDES encryption with the addition of HMACSHA–1 for integrity checking[21]. TripleDES is a symmetric encryption algorithm that utilizes an older sibling called Data Encryption Standard or DES. DES, in turn, is a block cipher that utilizes a 56 bit key on each 64 bit chunk of data. TripeDES improves the features of DES considerably by using the DES cipher three times with three distinct keys, thus yielding a 168 bit key. A discussion of the aspects and functioning of either symmetric or asymmetric encryption, Public Key Encryption, *etc.* is beyond the scope of this Technical Brief; however there are references in the Bibliography that developers can consult.

Password Extraction

The use of so–called "password crackers" has been a challenge for FileMaker Pro developers seeking to protect their intellectual property and to assure data integrity and data confidentiality in solutions. These nefarious little tools simply extracted the passwords from the file and revealed them in clear text. Additionally, as I mentioned earlier, weakly encoded passwords passing in TCP packets across a network were susceptible to interception and decoding.

FileMaker Pro 7 does not store passwords in the database file. Instead it stores a *hash* of the password. A *hash* is the one–way, non–reversible result of performing a mathematical rule on a string of data. Even if the hash[22] 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. This makes it extremely difficult to "crack" passwords.

Other Issues Addressed

FileMaker Pro 7 employs a Unicode text format. Temporary files sent to client workstations arrive in a compressed Unicode format, making their reading by text editor very difficult. In addition to this compression, the files are strongly encoded.

As I discussed in the section on *Granularity,* FileMaker Pro 7 addresses the problems many solution developers, particularly commercial solution developers, had in earlier versions of striking a workable balance between protecting their intellectual property on the one hand and making their solutions reasonably customizable by end users on the other. The ability to give classes of users the ability to create new objects such as layouts, scripts, and value lists without affecting existing instances of those objects provides developers with far greater flexibility in the design of their solutions than was the case heretofore. This has considerable impact on their business models as well, as we will see in the final section of this Technical Brief.

Conversion Issues From Earlier Versions

Many developers and IS/IT/DBA managers will decide to convert their existing solutions from FileMaker Pro 6 or even earlier versions to take advantage of the many new features present in FileMaker Pro 7, FileMaker Server 7, and FileMaker Server 7 Advanced. Depending on how the security schema in these earlier versions is structured a number of previously used techniques will have to be abandoned, be restructured pre–conversion, or be ameliorated post–conversion if the security features are to yield comparable results to what they did previously. Conversion is a complex topic and there are numerous nuances to the security features alone. Developers should refer to additional documentation available on the FileMaker, Inc. web site for even more extensive information on these topics.

Mal–formed security schema in FileMaker Pro 6 files will be a particular source of difficulty on conversion. Lack of uniqueness for groups, inattention to case sensitivity of both passwords and groups, and assigning passwords to more than one Group—especially Groups with dissimilar access privileges—can produce unexpected results in converted files.

FileMaker Pro 6 Groups will convert to Privilege Sets in FileMaker Pro 7, and the conversion process will seek to duplicate the old privileges of Groups and their associated passwords as faithfully as possible. Nevertheless, developers should check the converted privileges and bear in mind that some items, such as sharing a file, now require Extended Privileges to be enabled. If a developer had created a Group in FileMaker Pro 6 that was solely for the "master password", e.g. a group called *"Developer_Only"* or similar nomenclature, the passwords for that Group will become accounts attached to the default [Full Access] Privilege Set in FileMaker Pro 7. Additionally, FileMaker Pro 7, as part of a broader effort to eliminate redundancy and reduce file detritus will consolidate Groups with identical privileges into a *single, unified* subordinate Privilege Set, usually with multiple account names and passwords attached. On conversion to FileMaker Pro 7 the old password from FileMaker Pro 6 becomes ***both*** the account name and the account password. Since account names are entered in the clear at log–on, this will reveal passwords. So, as a first step, developers should change the account name to something other than the password.

As a result of this process, there will be issues with conditional tests that relied on the old Group name in FileMaker Pro 6 as revealed by the Status(CurrentGroups) function. Status(CurrentGroups) now becomes Get(PrivilegeSetName) as one of the new Get Functions replacing Status Functions. The result of this test will be ***different*** in FileMaker Pro 7 than it was in FileMaker Pro 6 if the Privilege Set name is different than was the old Group Name. That occurs with the [Full Access] Privilege Set converted from whatever the developer had named the Group particular to just the "master passwords," and it also occurs where FileMaker Pro 6 Groups and passwords with *identical* privileges, but with *different* passwords, were consolidated.

Thus, for example, a ScriptMaker script step syntax based on the "master password" that said:

[If (PatternCount, Status(CurrentGroups), "Developer_Only"))]

that evaluated to True in FileMaker Pro 6 may now fail in FileMaker Pro 7 because it now reads [PatternCount (Get(PrivilegeSetName) ; "Developer_Only")] and the Privilege Set name is now [Full Access]. Similarly, in a situation where passwords with identical privileges have been individually mapped to different, identical Groups, a test that read, for example:

[If (PatternCount, Status(CurrentGroups), "SalesMgr"))]

that evaluated to True in FileMaker Pro 6 may now fail in FileMaker Pro 7 because the Group "SalesMgr" has been consolidated along with such Groups as "MarketingMgr" and "OperationsMgr" into a single Privilege Set named, for example, "MarketingMgr."

Developers must check converted files to identify and to ameliorate these anomalies. The following table lists many places where such tests might be found, although not necessarily all such instances.

Conditional Scripting [If…]	Calculation field formula
Record Level Access tests	Auto–entered calculated values
Field validations by calculations	Conditional value lists
Set Field and Insert Calculated Results ScriptMaker script steps	AppleScript or VB Script generated wholly or partially from calculated fields
Replace Function	Show Custom Dialog function

Developers will need to analyze their solutions for potential security schema issues before converting in many instances, using the DDR Tool found in FileMaker Pro 6 Developer or the MetaDataMagic and PasswordAdminsitrator tools from New Millennium Communications, Inc.[23] Check carefully for case sensitivity of passwords across all files of a solution. The passwords *Patrick Henry, Patrick henry,* and *PatrIcK henry*, all are identical in FileMaker Pro 6. They are distinct in FileMaker Pro 7.[24] MetaDataMagic can be used to identify locations where STATUS(CURRENTGROUPS) is used in order to attempt amelioration before conversion as well as after conversion.

Business Model and Operations Impact For Developers and IS/IT/DBA Managers

The new FileMaker Pro 7 and FileMaker Server 7 security features will have profound impact on the way that commercial solutions developers, consulting developers, and IS/IT/DBA managers all do their work. And the new features will significantly affect the business models that control the consulting practices of many developers.

Commercial solution developers for many years have had grave concerns about the security of their intellectual property. In order to provide maximum flexibility for end user customers, these developers have needed in many instances to distribute solutions in an unlocked or full access format. Alternatively they have spent a huge amount of time updating customer files, reimporting data, and managing passwords, Groups, and similar access issues.

In FileMaker Pro 7 the need for much of this extra work is eliminated. Developers can grant to end–users, usually administrators or *Superusers*, the ability to manage accounts and to create a range of objects including scripts, layouts, and value lists *without* affecting or having access to existing instances of these objects. Likewise, commercial developers will be freed of the concern that password crackers will extract "master" passwords from their solution files, thus exposing their work in an unintended fashion. Moreover, if the commercial solution developer wants to integrate his or her solution with an existing client solution, creating hooks that respond to calls from the existing client solution and protecting those hooks with custom Extended Privileges open an entirely new avenue for product design and deployment.

Consulting developers work in a different environment, developing customized solutions for specific client needs and business processes. The new security system facilitates that business model as well. Consulting developers can be freed of the responsibility for managing user accounts as organization personnel come and go or change their responsibilities and roles. Again, by empowering *Superusers* to create, disable, enable, delete, and reset accounts, the consulting developer can focus over the life of the project on perfecting the features of the system and creating database elements to respond to specific client business rules.

IS/IT/DBA personnel can now exploit existing assets to manage security for a wide range of FileMaker Pro based assets, both LAN/WAN, web browser–based, and third–party applications using ODBC or JDBC connectivity. Single sign–on authentication to these assets greatly enhances IT ability to fulfill overall organization security responsibilities and makes addition or removal of users a straightforward exercise. The introduction of encrypted data streams between FileMaker Server on one end and FileMaker Pro clients, including the Web Publishing Engine, on the other further assists IT in meeting organization security policy requirements.

Conclusion

The new FileMaker Pro 7, FileMaker Server 7, and FileMaker Server 7 Advanced security features offer an entirely new and dramatically stronger approach for intellectual property protection, data confidentiality, and data integrity. Through the ability to institute privilege control over FileMaker Pro objects at a fine granular level, to use industry standards based account authentication, and to provide encryption for data protection, the new security system allows developers and IS/IT/DBA managers much more security control and certainty than in the past. These features are available *ab initio* in newly constructed FileMaker Pro 7 files; and they are likewise immediately available when FileMaker Pro 7 converts a file from a prior version.

Developers and IS/IT/DBA managers must and should be serious about security. The new system gives them new tools to support that responsibility.

About the Author

Steven H. Blackwell is a Partner Member of the FileMaker Solutions Alliance and President and CEO of Management Counseling Services [http://www.FMP-Power.com]. A two–time winner of the FileMaker Excellence Award, he specializes in custom FileMaker Pro development, FileMaker Pro security consulting, and FileMaker Server deployment.

THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, AND FILEMAKER DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, OR THE WARRANTY OF NON-INFRINGEMENT. IN NO EVENT SHALL FILEMAKER OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES WHATSOEVER INCLUDING DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL, LOSS OF BUSINESS PROFITS, PUNITIVE OR SPECIAL DAMAGES, EVEN IF FILEMAKER OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY. FILEMAKER MAY MAKE CHANGES TO THIS DOCUMENT AT ANY TIME WITHOUT NOTICE. THIS DOCUMENT MAY BE OUT OF DATE AND FILEMAKER MAKES NO COMMITMENT TO UPDATE THIS INFORMATION.

(Endnotes)

[1] Bruce Schneier, founder of Counterpane Labs, the premier digital security firm, has eloquently and persuasively explained these issues at length in his book *Secrets & Lies Digital Security in a Networked World,* {New York, NY. John Wiley & Sons. 2000}.

[2] See FileMaker, Inc. Tech Info Letter Number 108462 and the *Web Security* White Paper available on the FileMaker, Inc. website for further discussion of these issues.

[3] A Domain Group from Active Directory or Open Directory, not an old FileMaker Pro 6 Group.

[4] Certain high ASCII non–alphanumeric characters may cause problems if used for access from web based accounts in FileMaker Pro 7. Consult the *Web Publishing Guide* PDF on the FileMaker, Inc. website for further details.

[5] Under the *File* menu, select File Options…-Open/Close and disable the automatic log–on.

[6] As defined by either Active Directory or Open Directory.

[7] An account whose Privilege Set grants full, unrestricted access to all parts of the file. This is somewhat analogous to the concept of the "master password" in FileMaker Pro 6. However, that term is now deprecated and inaccurate for use in FileMaker Pro 7.

[8] For which purpose there are numerous resources, some of which are noted in the Bibliography of this Technical Brief.

[9] The Account Name and password could be the same as the Privilege Set, e.g., *Marketing or Sales Manager.* Just be sure to delete these test accounts at the end of the development process. New actual accounts can then be assigned to each Privilege Set.

[10] See the January 2004 issue of *FileMaker Advisor* magazine for more information about role–based access.

[11] This function returns the name of the Account accessing the file. In the event of external authentication, it still returns the name of the account and **not** the Group name. See however the discussion on Authentication order.

[12] Converted databases that used auto–enter either the Creator Name or the Modifier Name in earlier versions may want to convert to the Account Name in the auto–entry options of the field definitions. However, if doing so, then the converted file must be adjusted either to make the Account Names match the User names or *vice versa*, especially if Record Level Access tests depend on these data.

[13] The actual security issue here is that by allowing their creation, the developer also allows *modification* of developer created objects of the same class.

[14] See the FileMaker, Inc. Technical Brief on FileMaker Server 7 by Wim Decorte for more information on logging.

[15] A typology sometimes used in establishing accounts on Terminal Services servers.

[16] Again, enterprise Domain Groups from Active Directory or Open Directory, **not** old FileMaker Pro 6 Groups.

[17] Including placing both the WPE and IIS/Apache on the same CPU, using a VPN between them if they are on separate CPU's as they likely may be, or creating a closed network between FileMaker Server 7 CPU, the WPE CPU, and the Apache/IIS CPU. The multi–homing capabilities of FileMaker Server 7 enhance the ability to create these configurations. See both the FileMaker Server and FileMaker Web Publishing Technical Briefs. Some preliminary Best Practices would seem to require that the WPE and IIS/Apache be run on the same CPU and that firewalls be used to assure data confidentiality. As an aside, dual processor CPU's are particularly well suited for these configurations.

[18] See the FileMaker, Inc. Technical Brief on FileMaker 7 Web Publishing by Cris Ippolite for more details.

[19] Each subscript called must also have this option toggled if it needs the ability to execute an action requiring [Full Access] privileges.

[20] This also means they are not visible in the User Interface. They are obscured when entered; they remain obscured.

[21] See http://java.sun.com/j2se/1.4.2/docs/guide/security/jce/JCERefGuide.htm for a good summary description of hashed message authentication code (HMAC).

[22] See http://searchDatabase.techtarget.com/sDefinition/0,,sid13_gci212230,00.html for more information on hashes.

[23] http://www.newmillennium.com

[24] And do not forget that users may have to be retrained from old habits of case insensitivity when entering passwords.

Bibliography

Books

Alberts, Christopher J. and Dorofee, Audrey J. *Managing Security Risks The OCTAVE™ Approach* (Addison-Wesley, New York, NY, 2002)

Barrett, Diane; Hausman, Kirk, and Weiss, Martin. *Security+* (Que, Indianapolis, IN, 2003)

Schneier, Bruce. *Secrets & Lies Digital Security in a Networked World* (John Wiley & Sons, New York, NY, 2000)

Singh, Simon. *The Code Book* (Anchor Books, New York, 1999)

Strebe, Matthew *Network Security Jumpstart* (Sybex, San Francisco, 2002)

Articles

"Internet Security" *Time,* 7/2/2001

Andress, Mandy, and Edward, Mark T. "Beware Wireless Security Woes" *E Business Advisor* March 2002

Chang, Stephanie and Janowski, Davis D. "The lay of the wireless LAN" *PC Magazine*, 5/21/2002

Hawkins, Dana. "Hide and they can't seek" *US News & World Report* 5/19/2003

Kerstetter Jim and Weintraub, Arlene. "Cyber Alert Portrait of an Ex-Hacker" *Business Week*, 6/9/2003

Kerstetter Jim. "You're Only As Good As Your Password" *Business Week*, 9/2/2002

Marelia, Darren. "AD network Interactions" *Windows & .Net magazine*, 3/1/2003

Vacca, John R. "Save Money With a Secure Remote-Access VPN" *Business Security Advisor,* July/August 2002