FileMaker and HIPAA—A Tool of Compliance

DISCLAIMER: FileMaker—A Tool of Compliance was written for FileMaker developers to provide a basic understanding of 45 CFR Part 164 and 21 CFR Part 11. This document is not intended to be a substitute for legal consultation. This article expresses the author's opinion and understanding of the regulations and does not claim to give definitive or comprehensive answers or the 'right' interpretation to many of the complex and often ambiguous questions which are brought up by the new HIPAA regulations. The author recommends that prior to undertaking HIPAA- or Part 11-related projects that developers review the appropriate regulations and other appropriate corroborations.

Part I "A Discussion of Regulations and Roles"

The Beginning of One Developer's Tale

"Is FileMaker compliant?"

"Can you make our system compliant?"

> *In the last few years, there has been a steady increase in the number of questions relating to FileMaker and regulations such as HIPAA, Part 11, and Section 508. As a firm believer that FileMaker can do just about anything, when a client asked about HIPAA compliance, I was sure the answer would be a resounding "Why yes, of course." Still, I didn't know for sure. I needed to learn more about the FDA regulations governing electronic records and electronic signatures (21 CFR[1] Part 11). As a pharmaceutical manufacturer, my inquiring client was well acquainted with the FDA's requirements. What they were uncertain of was whether FileMaker was up to the challenge.*

FileMaker Pro's robust feature set provides a powerful tool of compliance with which you can indeed meet regulatory requirements. This is important because in-house developers and independent consultants alike are likely to hear similar questions with increasing regularity as our growing dependence on digital information demands oversight to ensure integrity and protect against unauthorized access.

To achieve compliance, you must first understand what it means to be "compliant." This paper will focus on meeting the requirements of Part 11 and HIPAA regulations[2], and the developer's role in achieving compliance. In addition to detailing specific technical requirements, the regulatory intent of each will be discussed to augment your understanding and guide your development choices.

What exactly are the Part 11 & HIPAA regulations?

21 CFR Part 11—Electronic Records; Electronic Signatures

At first glance, the scope of Part 11 is fairly simple. The regulations merely establish the criteria under which electronic records and electronic signatures are considered by the FDA to be legal equivalents to paper records and handwritten signatures. Part 11 applies to electronic records that are created, modified, maintained, archived, retrieved, or transmitted under any FDA records requirement. Part 11 also applies to electronic records submitted to the FDA under the requirements of the Public Health Service Act and the Federal Food, Drug, and Cosmetic Act.

To ensure that a signer cannot reject their electronic records or signatures as invalid, procedures and controls must be in place to ensure authenticity, integrity, and, in some cases, confidentiality. To ensure the legitimacy of electronic records and signatures, systems must be validated to ensure accuracy, reliability, consistency of intended performance, and ability to discern invalid or altered records.

[1] *CFR refers to the Code of Federal Regulations. CFR Title 21 covers Food and Drugs, while CFR Title 45 covers Public Welfare. 21 CFR Part 11 (referenced in this document as "Part 11"); "Electronic records; electronic signatures" Both Titles 21 and 45 include regulations controlled by the Department of Health and Human Services (DHHS), however, Part 11 of Title 21 is specific to the Food and Drug Administration (FDA).*

[2] *The Health Insurance Portability and Accountability Act, 45 CFR, Part 164 (referenced as the "Security Rule")*

The regulations are brief, but the implementation is not. The validation process itself is an arduous adventure—not for the impatient or disorganized. For the developer working on a Part 11-compliant system, it will be helpful to keep in mind that your contributions toward obtaining FDA validation of the completed system may be as critical as your role in providing its core functionality.

45 CFR Parts 160, 162 & 164—HIPAA

Most anyone who's been to a medical facility lately has heard of HIPAA (often misspelled "HIPPA"), but few understand its meaning or impact. So what exactly does health insurance portability and accountability have to do with FileMaker? Nothing directly, but systems developed using FileMaker may fall under the HIPAA umbrella. So if you use or develop FileMaker-based solutions intended for use within the medical industry, you need to know about HIPAA.

In their entirety, the HIPAA regulations are sizable. The focus of this document is to assist you in meeting the requirements of a very specific portion of HIPAA known as the Security Rule (45 CFR Part 164). A brief review of the overall rule—while not essential to development—will enable you to intelligently converse on the topic with colleagues and clients.

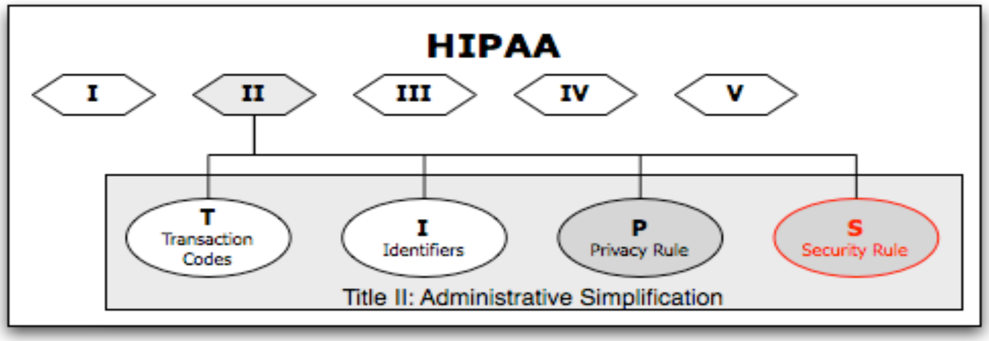


Figure 1—Title II, Administrative Simplification

HIPAA is an extensive set of regulations enacted by Congress in 1996 and administered by the U.S. Department of Health and Human Services. HIPAA consists of five titles. Title I protects health insurance coverage for workers and their families when they change or lose their jobs (hence the moniker). Titles III thru V[3] are not relevant to our discussion of compliance.

Title II, *Administrative Simplification*, consists of four separate provisions: Transaction Codes, Identifiers, Privacy, and Security. Transaction Codes and Identifiers define universal codes pertinent to billing systems. As a protector of patient privacy, and having gone into effect earlier, the Privacy Rule garnered more publicity than the fourth provision, but it has relatively little to do with system compliance.

The Security Rule is a developer's primary focus since it pertains specifically to the protection of electronic information.

When a covered entity (e.g., a medical practice) refers to HIPAA compliance, they may be referring to any of the four provisions within Title II: Administrative Simplification. Unless otherwise noted within this document, when speaking of HIPAA compliance in the context of systems or development, we are referring to the Security Rule.

Why examine HIPAA and Part 11 in tandem?

If your particular interest is HIPAA, you are probably wondering why you should care about Part 11. The two regulations appear to have little in common, so why discuss them together? Before answering this question, let's first clarify some key differences.

1. The HIPAA Security Rule is more extensive than Part 11, in both content and scope.
2. Lacking the specificity seen in Part 11, the Security Rule can be confusing with its use of three specification categories, two requirement levels, and guidelines that are vague by design.

[3] *Titles III thru V—Tax-related Health Provisions, Application and Enforcement of Group Health Plan Requirements, and Revenue Offsets*

3. There is no HIPAA counterpart to the FDA validation process, so confirming HIPAA compliance is less onerous. Unfortunately, it is also less precise, and the relative success of your efforts may be subject to interpretation.
4. The primary goal of Part 11 is to ensure the irrefutability of electronic records and electronic signatures, while the primary goal behind HIPAA's administrative simplification was cost savings.
   a. Improving the efficiency and effectiveness of our health care system through standardization saves money.
   b. Encouraging widespread use of electronic data interchange (EDI) is a major element of that homogeny.

The FDA sought to promote efficiency by allowing the submission of documents as electronic records with electronic signatures. In the absence of specifically defined policies to ensure their validity, these electronic submissions could be easily refuted—an undesirable possibility in any environment, and certainly unacceptable within the context of stringently controlled FDA submissions. The concise but explicit provisions of Part 11 were enacted to provide efficiencies of progress with the balance of protection.

In providing guidelines to ensure the validity and security of electronic data, the Security Rule seeks to promote similar efficiencies by encouraging the use of electronic data within the health care industry. Unlike Part 11, which applies to a relatively narrow sector, HIPAA's requirements were expected to affect more than a half-million health care establishments. Mandating strict and specifically defined requirements within the Security Rule presented several potential issues—including the probability that requirements would become quickly outdated as technology progressed. The specifics of HIPAA's Security Rule are therefore more vague than Part 11. All differences aside, the specific development techniques employed to meet the requirements of Part 11 may be similarly applied to satisfy elements of HIPAA.

A compliance proviso known as "reasonable and appropriate" allows for case-by-case evaluations, resulting in a broad range of compliance options. You will soon learn that not all HIPAA requirements are actually required—unless they are required. However, with FileMaker 8 there is no need to cut corners or settle for the minimum. Meeting all of the Security Rule's guidelines is not only possible, it is both reasonable and appropriate.

Why are HIPAA and Part 11 important to you?

If you work within an FDA-controlled environment or the health care industry, the integrity, confidentiality, and availability of the data within your control is sacrosanct. If you are an independent FileMaker developer, you may not be aware that the health care market represents over 560,000 businesses—of which more than 400,000 are doctor's offices and clinics. Not only is the market considerable, its potential need for FileMaker and your services is on the rise as the tides of regulatory and industry influence swell.

The Security Rule regulates all instances of electronic protected health information (ePHI) within the medical industry. This makes HIPAA compliance a concern for both existing and future FileMaker systems—regardless of size. According to a study by Forrester Research in 2004, 85% of U.S. doctors still rely on paper medical records. So what's the impetus for increased interest in FileMaker? Electronic Medical Records (EMRs). In his 2004 State of the Union Address, President George W. Bush formally christened the EMR bandwagon by setting a national goal that a majority of Americans have electronic medical records by 2014. With the federal government's appropriation of $42.5 million in 2005 to fund the Office of the National Coordinator for Health Information Technology, the EMR revolution began in earnest.

The American College of Physicians estimates that the health care industry could save as much as $30 billion per year if all patients' health records were in a digital format. An estimated 90% of medical practices already use Practice Management Software (PMS) to handle the administrative aspects of their business. Unlike the relatively consistent feature sets offered by PMS vendors, however, the requirements for EMR applications will be as diverse and unique as the specialties using them. FileMaker products are a natural choice for medical practices seeking to reap the benefits of EMR technology. Because FileMaker is so flexible, easy to use, and offers a very cost-effective development cycle, primary care physicians (PCPs), oncologists, cardiologists, mental health practitioners, and dentists alike can enjoy the advantages of EMR.

The Developer's Role

The developer's role—Part 11 compliance

Regardless of a system's size, obtaining FDA validation will most certainly require a team effort. In addition to the system's developer(s), the validation team will probably include subject experts, validation experts, QA staff, and one or more technical writers. Unlike other projects where your role as developer may be as chief architect leading the effort, your role in the validation process must also be supportive. In addition to meeting the technical requirements specified by Part 11, you will also be expected to support the validation team in other ways. Be prepared to serve as the FileMaker expert and champion, translator, advisor, and sounding board. Prior Part 11 expertise is not essential to achieving validation. However, a proven ability to successfully implement the security schema in FileMaker, and experience with scripted navigation controls are vital. Although audit controls have never been easier to implement, prior experience with audit trails will also prove invaluable. Since Part 11 specifically requires the developer of a system to have the education, training, and experience to perform their assigned task, FileMaker certification will be a vital asset—to yourself in securing the project and to your client in achieving validation.

If you plan to work on a system with Part 11 compliance requirements, ensure that your estimates provide for the requisite support, consulting, and documentation. Even if a team of technical writers is employed, validation will ultimately require details that only you can provide. Also keep in mind that your attendance at meetings not directly related to your development efforts is likely to be more frequent than with other projects. Moreover, total autonomy in your efforts is unlikely. The satisfaction of achieving validation is significant, but there is little room for ego.

The developer's role—HIPAA compliance

Meeting the unique data requirements of a cardiologist who wishes to share test results with referring PCPs won't pose a problem for the seasoned developer. However, to protect medical practices from the potential consequences of HIPAA non-compliance, HIPAA familiarity on the part of the developer is essential. Your familiarity with HIPAA will be equally critical if you are contracted as a consultant to assist an in-house developer with their compliance efforts.

In contrast to Part 11 compliance, which is formally validated by the FDA, the achievement of compliance with HIPAA's Security Rule is based on a covered entity's self-appraisal, which may be inaccurate. As developer or consultant, the covered entity will look to you for guidance. Your expertise should extend sufficiently beyond the technical aspects of compliance to clarify understanding and address common misconceptions. Your ability to recognize misinformation will enable you to provide more appropriate and valuable direction. You should know, for example, that it is not enough for your client to say, "so long as we only use the solution internally, we'll be compliant because we operate a HIPAA-compliant office." If your client says, "HIPAA doesn't apply to us because we don't submit our billing electronically," you should be able to counsel them on why and where it does apply.

With FileMaker as your development tool, addressing the technical points of HIPAA compliance may well be the easiest part of your job. As you'll soon learn, determining which details require your attention and agreeing on the scope of your effort may not be as simple.

If you are a covered entity, you should be confident that your developer has both sufficient understanding of the applicable regulations and the requisite skills to fulfill the technical role you require.

Is Compliance Complicated?

If compliance were simple, you would not be reading this document. Yes, compliance can be quite complicated, but you are likely to conclude that with FileMaker on your side, the actual programming is not as difficult as you might have expected.

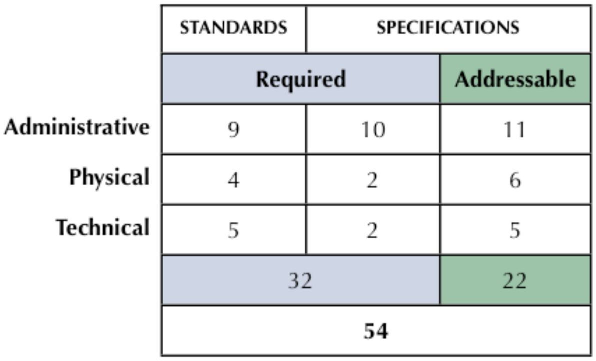
Complicated Compliance—Part 11 & FDA Validation

Incorporating the necessary functionality required for Part 11 compliance is arguably the easiest task. "Validation" is a process through which all system functions are documented and proven. This is an exhaustive undertaking that extends well beyond the average scope of development. It involves a system development lifecycle of assessments (gap-analysis); Current Good Manufacturing Practices (cGMP); Installation, Operational, and Performance Qualification (IQ, OQ, and PQ); Change Control management; auditing of software development and implementation processes; thorough documentation; training; and more.

Complicated Compliance—HIPAA

Standards and Implementation Specifications

The purpose of the Security Rule in HIPAA is to ensure the availability, confidentiality, and integrity of ePHI; and to protect against internal and external threats. The Security Rule consists of standards and implementation specifications[4], each of which is classified as either administrative, physical, or technical. The distinction between standards and implementation specifications is important because all standards are required, but implementation specifications may be either required or addressable. In each instance of an addressable item, a judgment call must be made by the covered entity as to whether or not it is reasonable and appropriate to implement. This case-by-case determination is commonly referred to as the "Reasonable & Appropriate" rule.



	STANDARDS	SPECIFICATIONS	
	Required		Addressable
Administrative	9	10	11
Physical	4	2	6
Technical	5	2	5
	32		22
	54		

Figure 2—Security Rule Organization by safeguard category, rule type, and requirement level

Your client may need assistance from you as they evaluate feasibility and implementation costs. While it may be determined that a majority of the guidelines will not directly impact your development effort, it is ultimately the covered entity's responsibility to make the compliance-related decision as to which guidelines will or will not be implemented. The following section discusses how this determination might be made and should help with efforts to assist your client in establishing a set of reasonable and appropriate specifications and expectations.

Reasonable & Appropriate

Assuming you are not by some miracle of genetic predisposition able to memorize the 54 individual standards and implementation specifications—including the distinction between required and addressable specification items—you will probably need to take some time to review each guideline individually to distinguish which are indeed required and applicable to the project at hand (see Figure 2). Given that only 12 of these are classified as technical, the scope of your effort may be to focus on these exclusively. In practice, you are more likely to find that your role also includes the implementation of features that are relevant to the other 42 guidelines.

For example, §164.308(a)(3)(ii)(B) *Workforce clearance procedure*—an addressable implementation specification within the Administrative safeguards—states that the covered entity must "implement procedures to determine that the access of a workforce member to electronic protected health information is appropriate." This may consist of a strictly non-technical determination as to who is permitted access to the system, in which case your contribution is unlikely. However, the business rules may dictate the logic for such a determination be incorporated within the system itself.

> *Consider the true case of an EMR shared by multiple mental health therapists. The general rule was to restrict visibility of a patient's data to their primary therapist and that individual's supervisor. This required the EMR to recognize both the patient's primary therapist and their supervisor. Simple enough to implement from a relational standpoint, but role-based data*

[4] *The Standards and Implementation Specifications may be referenced as "guidelines", "rules", "regulations", "requirements", "specifications", or "safeguards".*

> *restrictions prevented supervisors and therapists from accessing the same functionality. Still not so extraordinary, except when a therapist with her own patients was given additional responsibility as a supervisor of other therapists. This created a potential conflict between the user- and role-based privileges—requiring the EMR to determine if and when an individual had access to which data and which functionality.*

After making an item-by-item determination of which of the 54 guidelines apply to your project, you must then translate the applicable elements into specific requirements for the system.

Your first pass through the guidelines may be relatively painless, because the applicability of some will be fairly obvious. Where it becomes complex is in deciding which of the addressable items are both reasonable and appropriate to implement. The covered entity's decision not to implement an item must be based on the combined factors of a risk analysis, risk mitigation strategy, existing security measures, and cost.

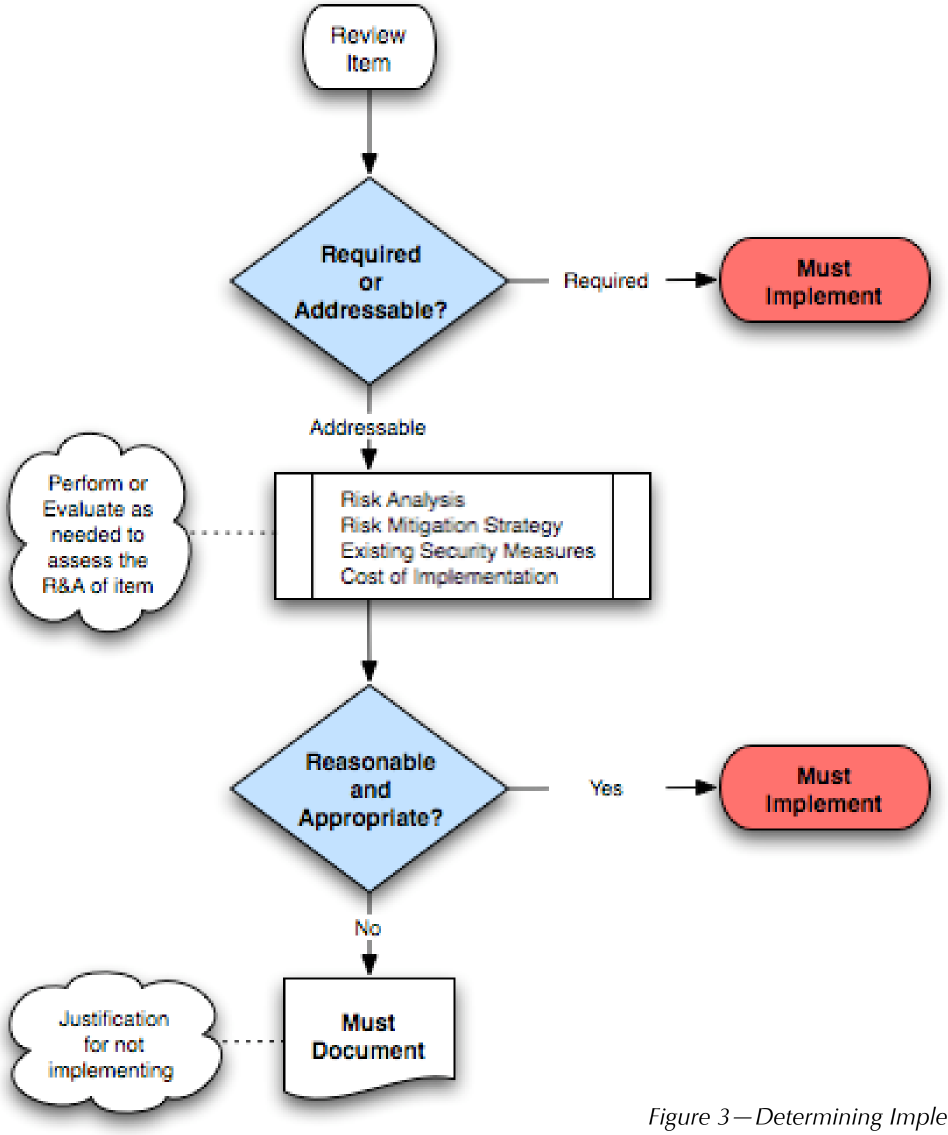


Figure 3—Determining Implementation

As referenced earlier, if your role as developer is consultative to a covered entity, your responsibility does not include making the final choice of which guidelines will be addressed within their solution. If you are an in-house developer, you may or may not be involved in the decision process. If you are the developer of a third-party application intended for use by covered entities, you will probably want to implement all guidelines within your control.

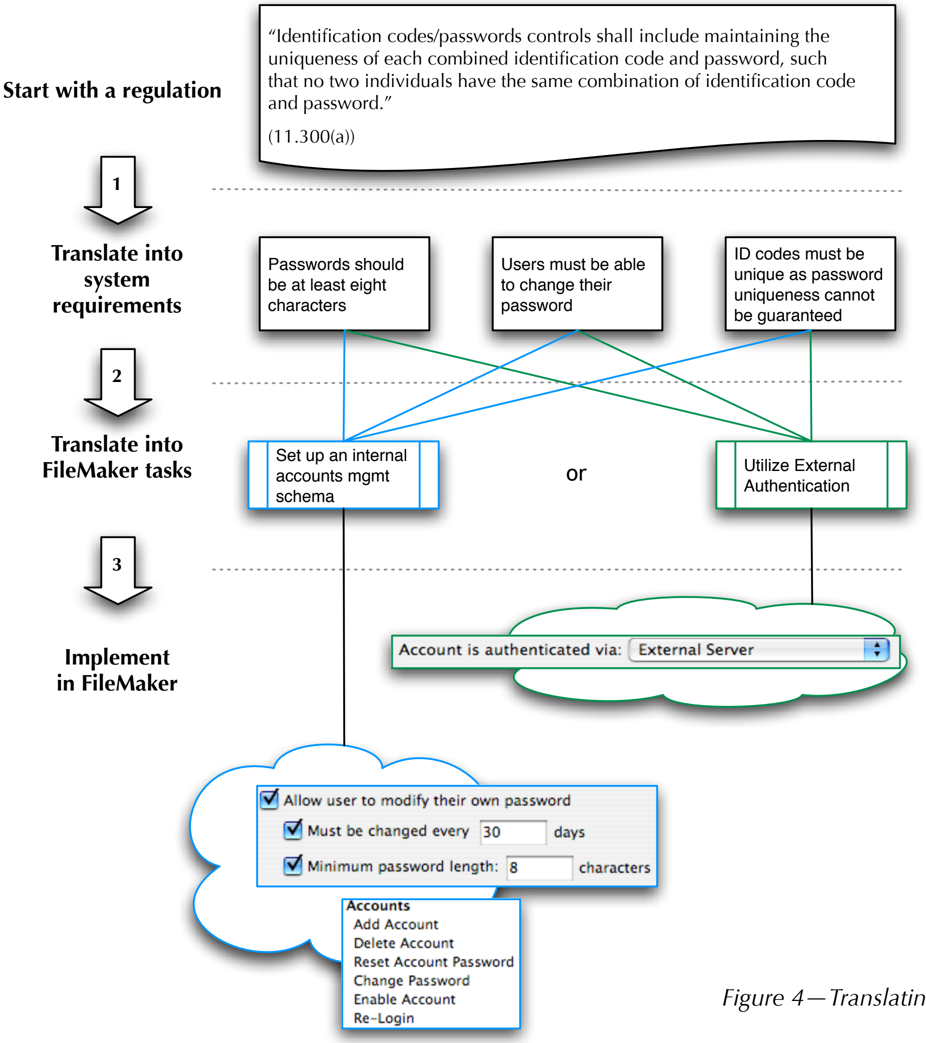
For each item, the decision maker must determine if it will be implemented. If an item is not addressed, the covered entity must document their justification (see Figure 3). This documentation is not the developer's responsibility, but your client may request documentation to support their choice.

Having thoroughly evaluated each item, the fruit of your labor would be a list of system requirements correlating to the original guidelines. Those requirements will then be translated into FileMaker-specific implementation items.

Translating regulations

The regulations themselves are the best starting point for determining what a system needs. The process of converting regulations into FileMaker-specific implementation tasks is the same whether addressing HIPAA or Part 11 requirements.

The regulations do not provide specific instructions, but they do express intent. Your first step is to convert this intent into system requirements. *Figure 4* demonstrates this first step with the listing of three requirements supporting a Part 11 mandate. When applicable and sufficient business rules already exist, such as minimum password length, your system requirements should be consistent. When implemented, these are expected to satisfy the intent of the regulation's text. Your requirements are not yet unique to FileMaker, nor do they indicate preference among the potentially numerous ways in which you might fulfill them. Accordingly, the next step is to translate your defined requirements into distinct FileMaker tasks.



"Identification codes/passwords controls shall include maintaining the uniqueness of each combined identification code and password, such that no two individuals have the same combination of identification code and password."

(11.300(a))

Passwords should be at least eight characters

Users must be able to change their password

ID codes must be unique as password uniqueness cannot be guaranteed

Account is authenticated via: External Server

☑ Allow user to modify their own password

☑ Must be changed every 30 days

☑ Minimum password length: 8 characters

Figure 4—Translating Regulations

In the example shown in Figure 4, there are at least two approaches identified: external authentication and an internal accounts management schema. With external authentication, all three requirements would be controlled outside of FileMaker. This may appear to be the simplest approach, but is it feasible to implement within the covered entity's environment? An alternative is to provide the necessary password management tools via schema and Accounts management functionality.

The regulations themselves rarely dictate a precise methodology for satisfying their intent, but you should make every effort to follow best practices in your design.

Resulting System Requirements

If you expect to develop a compliant system, a thorough review[5] of the applicable regulations is in order. In the interest of brevity the following short list summarizes the system requirements for Part 11 and HIPAA's security rule:

1. Audit log
2. Access reporting
3. Incident tracking
4. Access control
   a. validation
   b. granting
   c. role-based
   d. function-based
5. Contingency planning
6. User documentation
7. Auto log-off
8. Encryption
9. Data integrity
10. Data authentication
11. User authentication
12. Password management

The degree to which you must address each will depend entirely on the system, its environment, the regulations involved (Part 11 versus HIPAA), and the scope of your role as developer. We will discuss implementation of these following a review of specific regulation text.

To clarify, ePHI refers to individually identifiable health information that is transmitted by electronic media, or maintained in electronic media. Individually identifiable health information is a subset of health information, including demographic data that identifies the individual or with respect to which there is a reasonable basis to believe the information can be used to identify the individual. In other words, if an individual's identity can be reasonably ascertained based on the data, e.g., name, parent's name, phone number, etc., then that data is considered "individually identifiable" and therefore protected under the provisions of HIPAA. A list of identifiable data is provided in Part II of this document along with encryption implementation details.

Specific Regulatory Requirements—HIPAA Security Rule

The following section details each of the Security Rule's technical[6] guidelines. Followed by the author's comments, translation, and basic implementation recommendations (each of which is covered further in Part II, "*Implementation*").

Technical—Required

Each of the following items is required—unless it literally does not apply. For example, if a system's installation is limited to a stand-alone machine, it's possible that the Transmission Security requirement may not be required. If not required, the covered entity must document their justification for not implementing.

Access Control §164.312(a)(1)

Implement technical policies and procedures for electronic information systems that maintain electronic protected health information to allow access only to those persons or software programs that have been granted access rights as specified in §164.308(a)(4).

[5] *Links to the full text of Part 11 and the Security Rule are provided at the end of this document*

[6] *Administrative and physical guidelines are not detailed in this document.*

Policies and procedures are the responsibility of the covered entity. However, these must be supported by appropriate access controls within the software (serving to provide the procedures required by this provision).

Translation: *Define and follow processes which provide access only to authorized users (human or electronic).*

Implementation: *Deny access to unauthorized users. (Access Controls, User Authentication, and Password Management)*

Unique User Identification §164.312(a)(2)(i)

Assign a unique name and/or number for identifying and tracking user identity.

Translation: *A system must provide at least one unique identifier in order to distinguish individual users, i.e., no shared log-ins.*

Implementation: *Require user-specific log-ins. (Access Controls and User Authentication)*

Emergency Access Procedure §164.312(a)(2)(ii)

Establish (and implement as needed) procedures for obtaining necessary electronic protected health information during an emergency.

You may agree to address this as a joint effort, or the covered entity may prefer to address this procedurally without the developer's contribution.

Translation: *The covered entity must be able to retrieve their data during or immediately following an emergency which renders either the data or usual security controls unavailable.*

Implementation: *Follow reliable backup procedures, and develop a contingency plan appropriate to address the most likely emergencies.*

Audit Controls §164.312(b)

Implement hardware, software, and/or procedural mechanisms that record and examine activity in information systems that contain or use electronic protected health information.

The vague nature of this requirement allows for a broad range of possibilities. It fails to specify what activities must be recorded and examined. The logical presumption may be that an audit trail of all activity (new, edit, and delete actions) should be employed. Keeping in mind the overriding intent to protect ePHI, it might be reasonable to expect that activities which do not impact ePHI need not be recorded. For example, it is probably not necessary to audit global fields used to filter portal data.

Translation: *You must record and provide a method for reviewing system activity.*

Implementation: *Track data modification—where appropriate and relevant—and system access, and provide a mechanism for review. (Audit log, Access reporting, Incident tracking)*

Integrity §164.312(c)(1)

Implement policies and procedures to protect electronic protected health information from improper alteration or destruction.

There is no definition provided for what is considered proper or who is authorized. Both are determinations that the covered entity will need to provide. In the case of a third-party application, the rules which you have applied should be clearly documented so the covered entity may reference them in their compliance documentation.

Translation: *Protect ePHI from unauthorized tampering.*

Implementation: *Create a controlled environment where create, edit, and delete functionality is limited to authorized users under "proper" circumstances. (Data Integrity via Role- and Function-based Access Controls, User Authentication, Password Management, and Audit Log)*

Person or Entity Authentication §164.312(d)

Implement procedures to verify that a person or entity seeking access to electronic protected health information is the one claimed.

> *The methodology for accomplishing this is not specified because regulations are expected to endure while technology is expected to continue evolving.*
>
> ***Translation:*** *The system must be able to validate the unique identity of each user seeking access—including non-human connections.*
>
> ***Implementation:*** *Incorporate Access Controls and User Authentication*

Transmission Security §164.312(e)(1)

Implement technical security measures to guard against unauthorized access to electronic protected health information that is being transmitted over an electronic communications network.

> *While mandating that transmitted data must be protected, this specific requirement is the likely culprit of a very common misunderstanding about who must comply with the Security Rule. Perhaps you've heard this yourself: "The Security Rule doesn't apply to our practice because we don't transmit patient data." There are two major fallacies with that statement. The first, of course, is that the inapplicability of a single guideline does not invalidate a covered entity's responsibility to comply with other guidelines. The second is the frequently presumed notion that by "electronic communications network", the requirement applies only to electronic billing or the Internet.*
>
> ***Translation:*** *Protect against unauthorized access during any and all network transmissions.*
>
> ***Implementation:*** *Incorporate Access Controls, Data-level Encryption, Network Encryption via FileMaker Server, and External Security Measures.*

Technical—Addressable

As addressable items, the following are not required if the covered entity deems them either unreasonable or inappropriate.

Automatic Logoff §164.312(a)(2)(iii)

Implement electronic procedures that terminate an electronic session after a predetermined time of inactivity.

> *This is easily addressed in the case of hosted files, but with a bit of creativity it can be addressed within local files as well.*
>
> ***Translation:*** *Idle sessions pose an unacceptable security risk which must be prevented.*
>
> ***Implementation:*** *Utilize idle limits via FileMaker Server or establish Auto Log-Off Controls.*

Encryption and Decryption §164.312(a)(2)(iv)

Implement a mechanism to encrypt and decrypt electronic protected health information.

> *Whether utilizing a plug-in, custom function, or another methodology of your own design, there are a few important points to remember about this item. First, there is no delineation between types of ePHI to which this applies. The intent is for all ePHI to be protected by the encryption/decryption requirement. Second, the specific inclusion of decryption protects the availability of data. In practice, this could mean that encryption keys employed during your development will need to be disclosed to your client. Another option may be escrow of the key(s). You can, of course, protect your own intellectual property, but not to the extent that it prevents the covered entity from accessing their ePHI.*
>
> ***Translation:*** *All ePHI must be protected with encryption, but not to the extent that is no longer accessible. A decryption mechanism must therefore be provided for the retrieval of encrypted ePHI.*
>
> ***Implementation:*** *Apply an Encryption/Decryption schema to all fields containing ePHI.*

Mechanism to Authenticate
Electronic Protected Health Information §164.312(d)(2)

Implement electronic mechanisms to corroborate that electronic protected health information has not been altered or destroyed in an unauthorized manner.

This appears at first glance to be redundant with §164.312(c)1—a point which no doubt contributes to the confusion surrounding compliance—but it is actually something else entirely. While §164.312(c)1 states that you must protect against unauthorized alteration or destruction, §164.312(d)2 requires you to substantiate that such unauthorized activity has not occurred. So how do you prove that something did not happen? An audit trail will enable you to document edits, but will it record deletions as well? You could develop an all-inclusive audit trail mechanism to record creation, editing, and deletion; or perhaps you'll choose to implement multiple mechanisms. Again, the methodology is not specified, only the intent—which in this case is the ability to prove that unauthorized alteration or destruction has not occurred.

Translation: *A system must be able to substantiate that it has not permitted unauthorized alteration or destruction of ePHI.*

Implementation: *Use an audit trail to prove that only authorized alterations and deletions have occurred, but the auditing of activity does not prevent unauthorized activity. A combination of measures must be employed to ensure the failure of unauthorized attempts to alter or destroy ePHI. (Access Reporting, Incident Tracking, Access Controls, Auto Log-off, Encryption, User Authentication, and Password Management)*

Integrity Controls §164.312(e)(2)(i)

Implement security measures to ensure that electronically transmitted electronic protected health information is not improperly modified without detection until disposed of.

With sufficient audit controls in place, all modifications can be detected by the system, but is this adequate for your implementation? Is it enough for the system to be aware of modifications, or might a literal notification be more appropriate? Will this apply universally, only to specific data, only under specific circumstances, or perhaps when a specific combination of conditions exists?

Translation: *Protect ePHI during transmission (regardless of to whom) such that modification during transmission will be detected.*

Implementation: *Utilize Encryption and Access Controls to protect ePHI, and employ Data Authentication measures to verify Data Integrity.*

Encryption §164.312(e)(2)(ii)

Implement a mechanism to encrypt electronic protected health information whenever deemed appropriate.

More confusion ensues. Didn't we just cover this in §164.312(a)(2)(iv)? It would seem so, but no, not exactly. If you read carefully you will notice the differences. First, this only references encryption. A clear indicator that the intent here is protection rather than availability. The second key difference is seen at the end: "whenever deemed appropriate." Also directed at protecting ePHI, this adds an important layer to the encryption schema. Specifically, when must ePHI be encrypted? Not to be confused with "which ePHI must be encrypted," addressing the question of when to encrypt will require you to carefully define your processes for adding and editing data. Are there any instances when a user can add or edit ePHI without it being encrypted? The answer should be "no," but there may be cases when the business rules dictate otherwise and encryption is therefore deemed inappropriate.

Translation: *Encrypt ePHI. You may edit and view ePHI in a pre-encryption or decrypted state, but do not store unencrypted ePHI.*

Implementation: *Encrypt ePHI using an appropriate encryption schema.*

Implementation of one or several requirements,
in the absence of other reasonable controls, is incomplete.

Only 42 More…

The 12 technical safeguards detailed here cover only one of the Security Rule's three guideline categories. As shown in Figure 2, an additional 42 guidelines are contained within the Administrative and Physical safeguards. Some may seem irrelevant or beyond the scope of your responsibility as a developer, but you should familiarize yourself with each of the regulations before embarking on your own compliance campaign.

Though not technical in nature, the following required Administrative safeguards are of particular relevance to the development of a HIPAA-compliant FileMaker system.

Business Associate Contracts and Other Arrangements §164.308(b)(1)
A covered entity, in accordance with §164.306, may permit a business associate to create, retrieve, maintain, or transmit electronic protected health information on the covered entity's behalf only if the covered entity obtains satisfactory assurances, in accordance with §164.314 (a) that the business associate will appropriately safeguard the information.

> *HIPAA provides for a covered entity's need to allow business associates access to ePHI, as in the case of a FileMaker developer working on an EMR. Business associates not otherwise defined by HIPAA are not subject to the penalties provided by HIPAA; however, they must provide assurances of their willingness and ability to appropriately safeguard the ePHI.*
>
> ***Translation:*** *A covered entity may provide access to a business associate only after obtaining satisfactory assurances that appropriate safeguards will be employed.*
>
> ***Implementation:*** *If you are a covered entity, you must obtain satisfactory assurance in the form of a written contract prior to allowing a business associate to access the PHI for which you are responsible (see §164.308(b)(4) next).*
>
> *If you are a developer, be prepared to offer the necessary assurances. Regardless of how many projects you may have been entrusted with in the past containing sensitive data, do not expect or suggest that a medical client should give you a copy of their system. If it is necessary to review a potential client's system prior to executing an agreement, you should request that ePHI be removed from the system first. Should you find yourself in receipt of a file containing ePHI, you should delete the ePHI (but only after confirming that you were not inadvertently provided with the "only" copy). If the presence of data is essential to your preliminary review, you can substitute bogus data.*

Written Contract or Other Arrangement §164.308(b)(4)
Document the satisfactory assurances required by paragraph (b)(1) of this section through a written contract or other arrangement with the business associate that meets the applicable requirements of §164.308(a).

> *Commonly referred to as the Business Associate Agreement or Business Associate Contract (BAC), the execution of such a contract provides the covered entity with a mechanism for working with others who may have legitimate cause to access their PHI. Execution of a BAC does not transfer the burden of HIPAA compliance from the covered entity to their business associate. Nor does it impose HIPAA-defined penalties upon the business associate (unless the business associate is also a covered entity). However, a properly written BAC does create a contractual obligation for which the business associate is responsible.*
>
> ***Translation:*** *If a covered entity enters into an agreement with a business associate under the provisions of §164.308(b)(1), that agreement must be in writing and it must include certain provisions as defined by §164.314(a)(2)(i) or §164.314(a)(2)(ii).*
>
> ***Implementation:*** *You may obtain a sample BAC from the covered entity, a colleague, online resources, or you might prefer to review the applicable provisions and write your own. Whatever your source, the BAC is a binding contract so the proviso to seek legal advice applies. Remember also that even if a casual agreement is perfectly acceptable to you as the developer, the absence of an appropriate contract will put the covered entity in violation (if PHI is exposed).*

FileMaker—A Tool of Compliance

Part II "Implementation"

The Developer's Tale Continues

> *The pharmaceutical company was already using FileMaker—in fact, they were hoping to obtain FDA validation for a FileMaker system already in production. Believing I would find myself preaching to the choir, I entered the room with great confidence. I was surprised with the level of suspicion that usually begins with an unspoken, "surely you don't expect us to believe it's possible to…," as their FDA consultant proceeded to ask, "Can FileMaker…?"*
>
> *Not wanting to overstate or misrepresent capabilities—either mine or what FileMaker was capable of—I reflected carefully before responding. I pondered each question, and evaluated the specific techniques I could employ to achieve the requested results. Still, it seemed to take little more than a nanosecond of mental review to consistently reach the originally anticipated conclusion. "Why yes, of course," I responded, twenty-six times.*

"How will you make our system compliant?"

Before implementing features and safeguards with the expectation of bringing a system into compliance, you must first understand the regulations with which you need to comply. Next you should be able to translate those same regulations into non-FileMaker-specific system requirements. The core requirements referenced in the following discussions apply to both the Security Rule and Part 11 unless noted otherwise.

I have referred to FileMaker as a "Tool of Compliance" for some time, and recent releases have introduced features that enhance and truly simplify the effort. For those who are unfamiliar with the FileMaker family of products, those who may not be familiar with newer versions, and those who are not yet familiar with compliance issues, I'd like to recap some of my favorite compliance-supportive features.

Specific features that make FileMaker a "Tool of Compliance"

In addition to the factors of rapid application development (RAD), ease of use, and overall cost effectiveness, the FileMaker 8 family of products also provides several features that you will find invaluable contributors to your compliance efforts.

1. **Security Schema**
   The design of your security schema is essential to protecting your system from both internal and external risks.
   a. **External Authentication**
      External authentication, available via FileMaker Server, brings the advantages of Single Sign-On (SSO) to your system. There is little need to reinvent the user-password-expirations-and-privileges wheel when an existing user authentication mechanism is already in place. External authentication uses industry standards Active Directory and Open Directory.
   b. **Record-level Access (RLA)**
      The granularity of RLA controls provides a valuable tool for implementing user-specific, role-based, and situational access restrictions.
   c. **Extended Privileges**
      Further enhances the level and flexibility of control you can apply.

d. **Encrypted Client-server Traffic**
This feature—available when hosting files via FileMaker Server—should be considered an essential tool for most implementations. Using the SSL protocol, you can now encrypt the transfer of data between host and desktop clients.
IMPORTANT: Encrypted client-server traffic is not to be confused with encrypted ePHI.

2. **Account Management Tools**
The ability to script account-related actions allows you to provide extensive management capabilities without exposing and endangering a system with the otherwise unnecessary distribution of [Full Access] privileges. Scripted account management allows you to design the specific controls your system requires while returning password management to the end user—a mandatory feature for Part 11.

3. **Custom Menus**
Available with FileMaker Pro Advanced, this feature allows you to customize and improve the user experience while also restricting access to standard-but-potentially-dangerous features such as "Find and Replace"—or any other menu-based action that you need to restrict or control. Custom Menus also allow you to apply controlled routines to otherwise uncontrolled user activities such as New Record, Delete Record, Close Window, etc.—while providing users with convenient and familiar access to keyboard commands with which to trigger your custom actions. Multiple custom menus may be incorporated within a system to provide maximum control per user, per role, per layout, per script, etc. The possibilities are endless. (Note that FileMaker Pro Advanced is required to define Custom Menus, but once defined, Custom Menus are available to users with either FileMaker Pro or FileMaker Pro Advanced.)

4. **Tooltips**
An obvious enhancement to the user experience, "rollover" Tooltips may also contribute to documentation and training requirements. Tooltips can be defined to use validating calculations; allowing you to expose additional data to authorized users as needed, or to customize the information or message displayed based on user, role, situation, or other dynamic criteria. (FileMaker Pro Advanced is required to define Tooltips, but once defined, Tooltips are visible to users with FileMaker Pro, FileMaker Pro Advanced, or accessing a system via Instant Web Publishing.)

5. **Custom Functions**
You can utilize custom functions to protect (i.e., hide) key information, and to simplify some of the unavoidable complexities of working in a controlled environment. For example, frequently used, lengthy nested or external function calls can be reduced and combined within a custom function. Recursive functions can be performed via Custom Function. The security surrounding encryption is greatly enhanced with the use of Custom Functions, and the introduction of Custom Functions revolutionized the design of audit trails. (Note that FileMaker Pro Advanced is required to define Custom Functions, but once defined, Custom Functions are available to users with either FileMaker Pro or FileMaker Pro Advanced.)

6. **Variables**
In addition to reducing the clutter and management requirements associated with untold volumes of global fields, session- and global, script-based variables can be used to eliminate unnecessary exposure of protected information, and to control dynamic environment-, user-, and role-based access triggers.

7. **Script Parameters**
The ability to pass fixed and dynamic script parameters allows you to efficiently control access to script-driven functionality.

8. **Tab Controls**
There is little security advantage to using tab controls, but they offer significant interface enhancement with minimal development effort while substantially reducing layout management requirements (and requisite controls). A system which would otherwise require dozens of controlled interface layouts can now be reduced to a handful of accessible layouts.

Putting the Tools to Work

You've read and translated your regulations. You've confirmed the various relevant business rules with your client. Your tools are at hand, and your FileMaker database is waiting to be transformed. So what, exactly, do you do next?

Your "next step" may be quite different from someone else's, because the individual requirements of your systems will vary—sometimes significantly. From the system expected to meet the needs of a single practitioner, to the EMR shared by 100 users via remote access, to the maintenance management system with 350 users at a manufacturing plant, and the cardiology practice needing to share their test results with primary care physicians online. Your system's business requirements will likely dictate your next several steps more-so than compliance. That having been said, you should not move too far down the development path without a basic understanding of the compliance-related functionality your system will ultimately require.

With rare exception, the following is not intended to serve as a how-to tutorial on specific techniques that must be used in order to achieve compliance. FileMaker is a flexible tool, with which you can choose your most appropriate path. Here we will examine common factors, potential issues, and approaches for the implementation suggestions identified earlier.

Audit log

What exactly is an *audit log* (aka *audit trail*)? Wikipedia defines an audit trail as "a chronological sequence of audit records, each of which contains evidence directly pertaining to and resulting from the execution of a business process or system function." Webopedia defines an audit trail as "a record showing who has accessed a computer system and what operations he or she has performed during a given period of time." Industry-specific variations on the meaning can be found, but it would be safe to say that the audit trail represents an electronic cousin of the paper trail.

To meet the requirements for HIPAA or Part 11 compliance, the purpose of our audit trail is to record activity that modifies or deletes data. Our audit trail will not prevent unauthorized alteration or deletion of data. The intent is to record activity that will be examined regularly for inconsistencies or unauthorized activity. An audit trail can also be quite useful in troubleshooting certain data issues. If, for example, a client is claiming that data has "mysteriously" changed, a reliable audit trail will allow you to spot when, where, and by whom the data was altered. Perhaps a user did not intentionally edit the data, but a script they were running did. The information in an audit trail can be quite valuable, both to the developer and their client. Though a bit more complex to set up, and not required for HIPAA compliance, an audit trail can also be helpful in reverting data that should not have been modified.

So how does one set up one of these sensational tools? To answer that, let's back up for just a moment and review what we want to achieve. At minimum, our end result should provide us with the following detail for each modification:

1. Field name
2. Old data
3. New data
4. User name
5. Timestamp

If we were working with only a handful of data fields, we could maintain a history field for each data point, using an auto-entered calculation to build each field's activity history. Not a very practical approach for even the simplest of files. But we are only required by the Security Rule to track alterations and deletions to ePHI, so a simple approach may seem more appealing. In time, the simple approach will fail to live up to its promise of ease, and you will find yourself expanding the schema field by field, or layout by layout.

So how should a grown-up audit trail work? The specific mechanics are up to you, but ponder your requirements and options carefully before settling on your approach. First, explore the opportunity to harvest other valuable information while polling your modifications. Beyond the five data points already listed, you'll probably want to know the account and/or privilege set of the user (particularly useful when troubleshooting). Knowing the name of the script that was running at the time of an edit can also be helpful. In addition to the timestamp, it might be appropriate to capture an IP or MAC address. To document the context of an edit, you could identify the active layout. There are numerous dynamic tid-bits you might consider valuable, and most can be obtained with ease using FileMaker Get ( ) functions. After your laundry list of essential and desirable details has been assembled, you should specifically define what data will be audited, and possibly, under what circumstances modifications will or will not be tracked.

The Dynamic Audit Trail— No Plug-ins Required

> *I was advised by my client that obtaining FDA validation of Part 11 compliance would require an audit trail of all cGMP data[7]. Attempting to implement an overly simplistic auditing model against the volume of fields classified as cGMP would have been inefficient to deploy and cumbersome to manage. It was clear that a simple approach would ultimately prove to be anything but simple, and with hundreds of layouts across multiple files to consider, reliability would have been questionable.*
>
> *With over 350 system users, implementing a plug-in-dependent audit trail was not a viable option due to licensing costs. There had to be a better way—something that would work regardless of which fields were used, or where. Something more dynamic was needed to minimize long-term impact and development overhead.*

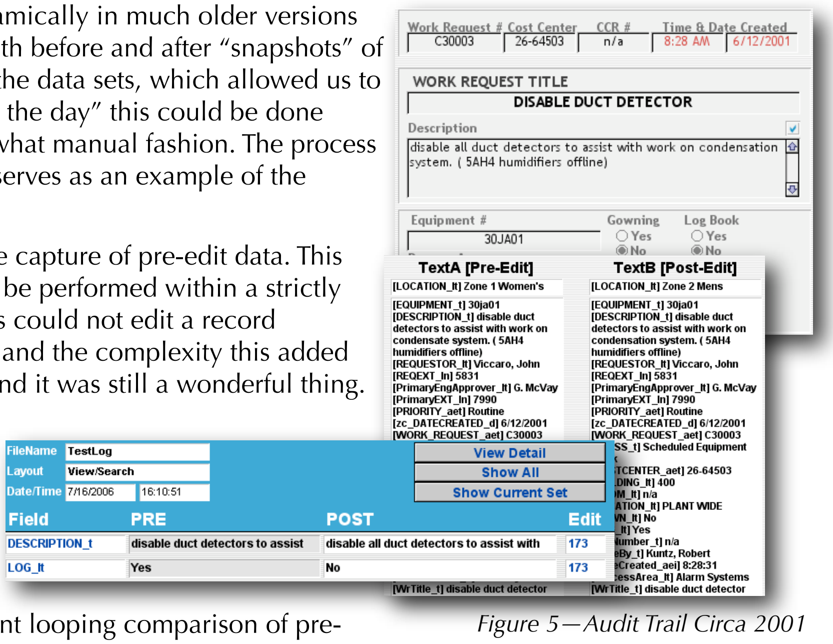
Inspired by the introduction of GetField in FileMaker 5.5, the principles described here have been successfully implemented in deployments of FileMaker versions 5.5 through 8.5. The specific programming details have evolved over time to take advantage of features that simplify the implementation requirements and enhance performance, but the fundamental process remain unchanged.

How it was…

In order to track changes dynamically in much older versions of FileMaker, we identified both before and after "snapshots" of our data. We then compared the data sets, which allowed us to identify the changes. "Back in the day" this could be done successfully, albeit in a somewhat manual fashion. The process was fairly easy to follow and serves as an example of the concept.

Recording an edit required the capture of pre-edit data. This meant that editing could only be performed within a strictly controlled environment. Users could not edit a record without entering "Edit Mode" and the complexity this added to a system was substantial. And it was still a wonderful thing.

Following the user's edit, a capture of post-edit data was required, so users were trained to "commit" their edits. The user's commit would trigger the capture of post-edit data and a subsequent looping comparison of pre- and post-data. Prior to the introduction of the Commit Record script step, allowing a user to cancel out of edit mode required the incorporation of a "revert" function.



Work Request #	Cost Center	CCR #	Time & Date Created	
C30003	26-64503	n/a	8:28 AM	6/12/2001

WORK REQUEST TITLE

DISABLE DUCT DETECTOR

Description ☑

disable all duct detectors to assist with work on condensation system. (5AH4 humidifiers offline)

Equipment # 30JA01

Gowning ○ Yes ● No

Log Book ○ Yes ● No

TextA [Pre-Edit]	TextB [Post-Edit]
[LOCATION_lt] Zone 1 Women's	[LOCATION_lt] Zone 2 Mens
[EQUIPMENT_t] 30ja01 [DESCRIPTION_t] disable duct detectors to assist with work on condensate system. (5AH4 humidifiers offline) [REQUESTOR_lt] Viccaro, John [REQEXT_ln] 5831 [PrimaryEngApprover_lt] G. McVay [PrimaryEXT_ln] 7990 [PRIORITY_aet] Routine [zc_DATECREATED_d] 6/12/2001 [WORK_REQUEST_aet] C30003 [WrTitle_t] disable duct detector	[EQUIPMENT_t] 30ja01 [DESCRIPTION_t] disable duct detectors to assist with work on condensation system. (5AH4 humidifiers offline) [REQUESTOR_lt] Viccaro, John [REQEXT_ln] 5831 [PrimaryEngApprover_lt] G. McVay [PrimaryEXT_ln] 7990 [PRIORITY_aet] Routine [zc_DATECREATED_d] 6/12/2001 [WORK_REQUEST_aet] C30003 SS_t] Scheduled Equipment k TCENTER_aet] 26-64503 DING_lt] 400 M_lt] n/a ATION_lt] PLANT WIDE VN_lt] No _lt] Yes Number_t] n/a eBy_t] Kuntz, Robert eCreated_aei] 8:28:31 cessArea_lt] Alarm Systems [WrTitle_t] disable duct detector

FileName: TestLog
Layout: View/Search
Date/Time: 7/16/2006 16:10:51

Field	PRE	POST	Edit
DESCRIPTION_t	disable duct detectors to assist	disable all duct detectors to assist with	173
LOG_lt	Yes	No	173

Figure 5—Audit Trail Circa 2001

The data captures were achieved using design functions to capture the current layout's field names, then to loop through those names and retrieve their contents.

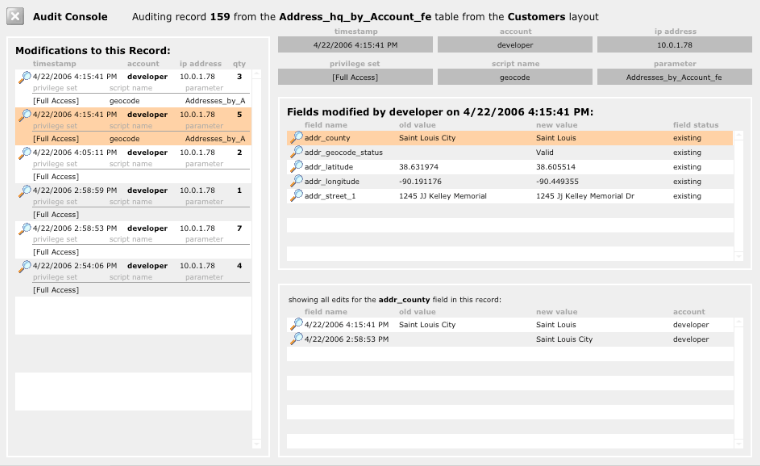
Figure 5 shows an edited screen, the pre/post edit comparison, and the resulting Audit Log entries. With its use of design functions, the audit trail was "dynamic" and included any desired field on any layout. New fields could be added to the system and to any layout, without the need to update

[7] *Current Good Manufacturing Practices (cGMP)*

anything to keep the audit trail functioning. This was very cool stuff.

How it is…

The same construct would work today. However, with the introduction of FileMaker 7 we were able to say good-bye to "edit mode" and scripted pre- and post-edit capture loops. We were also able to eliminate the overhead of storing separate log entries. With the introduction of FileMaker 8 and Variables, we were able to streamline the process even further.



Audit Console Auditing record **159** from the **Address_hq_by_Account_fe** table from the **Customers** layout

timestamp	account	ip address
4/22/2006 4:15:41 PM	developer	10.0.1.78

privilege set	script name	parameter
[Full Access]	geocode	Addresses_by_Account_fe

Modifications to this Record:

timestamp	account	ip address	qty	privilege set	script name	parameter
4/22/2006 4:15:41 PM	**developer**	10.0.1.78	**3**	[Full Access]	geocode	Addresses_by_A
4/22/2006 4:15:41 PM	**developer**	10.0.1.78	**5**	[Full Access]	geocode	Addresses_by_A
4/22/2006 4:05:11 PM	**developer**	10.0.1.78	**2**	[Full Access]		
4/22/2006 2:58:59 PM	**developer**	10.0.1.78	**1**	[Full Access]		
4/22/2006 2:58:53 PM	**developer**	10.0.1.78	**7**	[Full Access]		
4/22/2006 2:54:06 PM	**developer**	10.0.1.78	**4**	[Full Access]		

Fields modified by developer on 4/22/2006 4:15:41 PM:

field name	old value	new value	field status
addr_county	Saint Louis City	Saint Louis	existing
addr_geocode_status		Valid	existing
addr_latitude	38.631974	38.605514	existing
addr_longitude	-90.191176	-90.449355	existing
addr_street_1	1245 JJ Kelley Memorial	1245 Jj Kelley Memorial Dr	existing

showing all edits for the **addr_county** field in this record:

field name	old value	new value	account
4/22/2006 4:15:41 PM	Saint Louis City	Saint Louis	developer
4/22/2006 2:58:53 PM		Saint Louis City	developer

Figure 6—Audit Trail Circa 2006

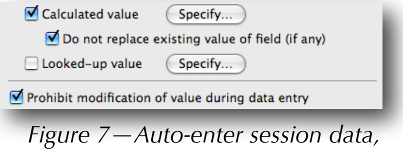
With recursive Custom Functions, auto-entered values, and XML, we are now able to generate comprehensive audit logs such as the one shown in Figure 6.

Although the cost factor of requiring a plug-in for 350 users was the impetus for building the 2001 audit trail shown in Figure 5, this is a valid approach to audit trail creation that may be appropriate for your needs. The audit trail is an essential component of a compliant system, so take care to explore your options thoroughly.

Access reporting

A mechanism for review of system access is necessary for compliance with several of the Security Rule guidelines. The format you choose for this Access Log will not impact compliance (i.e., onscreen view or printed report), so long as it is accessible to the appropriate individuals. In order for the covered entity to monitor activity and verify that only authorized individuals have been granted access to the system, your Access Log will need to include identifying information about the user in addition to the specifics of when access was provided. You may also choose to include additional details such as when access ended and other helpful details.

1. To ensure the integrity of your Access Log, users must not be permitted to modify the log data. Use auto-entered values that do not allow modification.

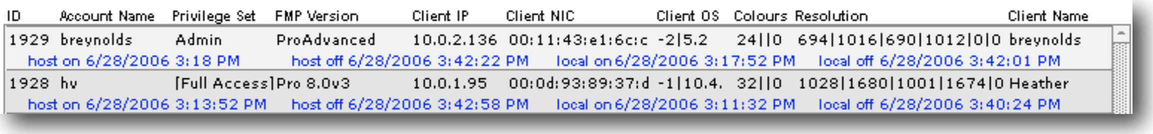


☑ Calculated value
☑ Do not replace existing value of field (if any)
☐ Looked-up value
☑ Prohibit modification of value during data entry

Figure 7—Auto-enter session data, prohibit modification

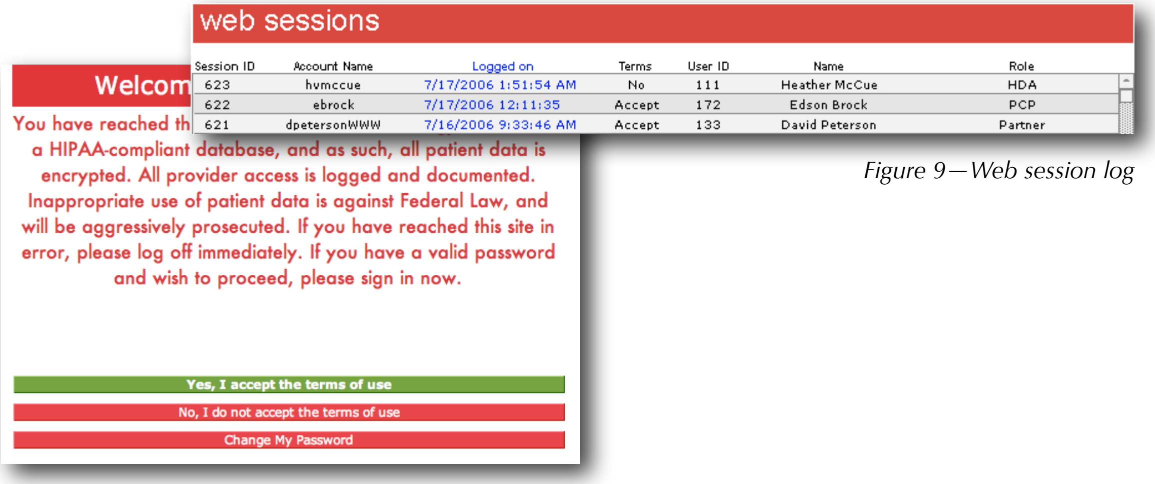
2. To ensure that a user's access is always reported, do not allow system entry without the creation of a session record. This can usually be accomplished within a system's "START" script. If your system consists of multiple files, you will need to determine how best to enforce session logging within the available structure.
3. If an end-of-session timestamp is also being recorded, insert this within a closing script.
4. If your system is hosted, include a host timestamp as well. This provides a consistent reference regardless of time zones or local, user-set system clocks.

   Note: When a user is disconnected by FileMaker Server due to an idle time out, the closing script does not run and their exit time will not be recorded.
5. In accordance with other requirements and preferences, you may need to record additional session detail or provide various reporting options. Following are a few examples of Access Logs and Activity Reports. Figure 8 is self explanatory. Figure 9, Web session log, documents access details including the user's acceptance or rejection of the stated Terms of use. If the user does not agree, they are logged out. We cannot identify a specific user without first authenticating them, so the Instant Web Publishing (IWP) visitor is taken to the Welcome page immediately following login. Their web session record has already been created and all that remains is to accept or reject the terms. Users may also be prompted to change their



```
ID    Account Name  Privilege Set  FMP Version    Client IP    Client NIC         Client OS  Colours Resolution                   Client Name
1929  breynolds     Admin          ProAdvanced    10.0.2.136   00:11:43:e1:6c:c  -2|5.2     24||0   694|1016|690|1012|0|0 breynolds
   host on 6/28/2006 3:18 PM     host off 6/28/2006 3:42:22 PM     local on 6/28/2006 3:17:52 PM     local off 6/28/2006 3:42:01 PM
1928  hv            [Full Access]Pro 8.0v3        10.0.1.95    00:0d:93:89:37:d  -1|10.4.   32||0   1028|1680|1001|1674|0 Heather
   host on 6/28/2006 3:13:52 PM  host off 6/28/2006 3:42:58 PM     local on 6/28/2006 3:11:32 PM     local off 6/28/2006 3:40:24 PM
```

Figure 8—Session log



web sessions

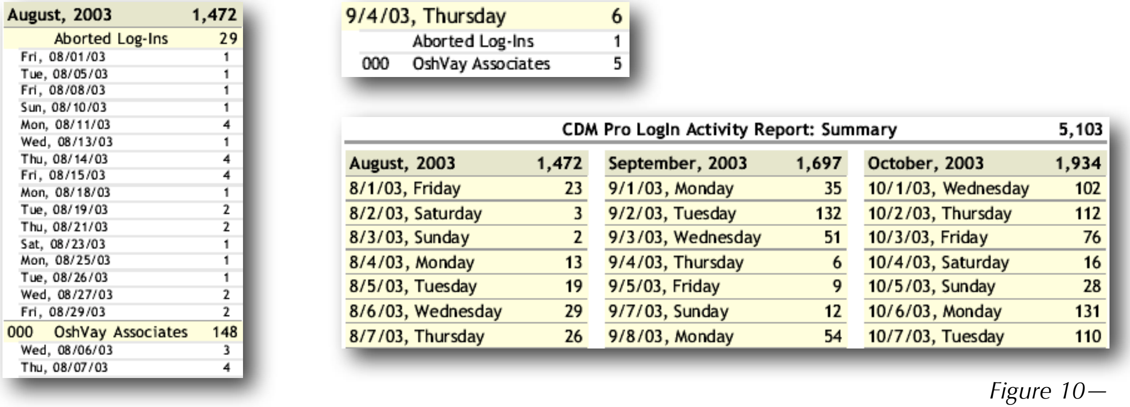
Session ID	Account Name	Logged on	Terms	User ID	Name	Role
623	hvmccue	7/17/2006 1:51:54 AM	No	111	Heather McCue	HDA
622	ebrock	7/17/2006 12:11:35	Accept	172	Edson Brock	PCP
621	dpetersonWWW	7/16/2006 9:33:46 AM	Accept	133	David Peterson	Partner

Welcom

You have reached th
a HIPAA-compliant database, and as such, all patient data is encrypted. All provider access is logged and documented. Inappropriate use of patient data is against Federal Law, and will be aggressively prosecuted. If you have reached this site in error, please log off immediately. If you have a valid password and wish to proceed, please sign in now.

Yes, I accept the terms of use
No, I do not accept the terms of use
Change My Password

Figure 9—Web session log



August, 2003	1,472
Aborted Log-Ins	29
Fri, 08/01/03	1
Tue, 08/05/03	1
Fri, 08/08/03	1
Sun, 08/10/03	1
Mon, 08/11/03	4
Wed, 08/13/03	1
Thu, 08/14/03	4
Fri, 08/15/03	4
Mon, 08/18/03	1
Tue, 08/19/03	2
Thu, 08/21/03	2
Sat, 08/23/03	1
Mon, 08/25/03	1
Tue, 08/26/03	1
Wed, 08/27/03	2
Fri, 08/29/03	2
000 OshVay Associates	148
Wed, 08/06/03	3
Thu, 08/07/03	4

9/4/03, Thursday	6
Aborted Log-Ins	1
000 OshVay Associates	5

CDM Pro LogIn Activity Report: Summary 5,103

August, 2003	1,472	September, 2003	1,697	October, 2003	1,934
8/1/03, Friday	23	9/1/03, Monday	35	10/1/03, Wednesday	102
8/2/03, Saturday	3	9/2/03, Tuesday	132	10/2/03, Thursday	112
8/3/03, Sunday	2	9/3/03, Wednesday	51	10/3/03, Friday	76
8/4/03, Monday	13	9/4/03, Thursday	6	10/4/03, Saturday	16
8/5/03, Tuesday	19	9/5/03, Friday	9	10/5/03, Sunday	28
8/6/03, Wednesday	29	9/7/03, Sunday	12	10/6/03, Monday	131
8/7/03, Thursday	26	9/8/03, Monday	54	10/7/03, Tuesday	110

Figure 10—
Printed Access Activity reports show daily logins, summarized by month & quarter;
per-user monthly logins, detailed by day; and a daily login summary, detailed by user

password, after which they are returned to the welcome page, i.e., they can not access the system data without first accepting the terms.

Incident tracking

Before you can track incidents, you must first define within the context of your system, which events will constitute a reportable incident. In Figure 9, a user's rejection of the stated terms of use has been recorded. In Figure 10, "Aborted Log-Ins" are tallied right along with valid, identified users. Login failures are among the more obvious incidents worthy of being tracked, but if you cannot identify the user without a successful authentication, how will you record the failure?

Using External Authentication via FileMaker Server, responsibility for initial access failures is effectively transferred to the domain server, and implementation of an audit trail provides a degree of incident tracking in that it will record authorized and unauthorized modifications alike. However, use of of these will not relieve you of the need to track other events that may represent risk.

If a user attempts to modify or delete ePHI for which they do not have modification privileges, the unauthorized attempt to alter would be considered a trackable incident. If your system provides a re-login feature, a failure to successfully log back in might be considered a trackable incident. If your system includes medical billing detail and controls are in place to limit the available code set, a user's attempt to enter an unauthorized code might be considered a trackable incident.

If your system allows a user to "sleep" their session without exiting, a failed authentication when attempting to awaken the session would probably be a trackable incident.

Once you have defined a specific event as a trackable incident, you will then need to decide how best to "track" it. If you choose to track certain incidents collectively in a log, who will have access to monitor it, and will they be alerted when something has been added?

In addition to documenting any identified incidents, appropriate response measures should also be followed, e.g., closing the session and exiting the application.

Access controls

The primary intent of the Security Rule is to protect ePHI from unauthorized access, alteration, or deletion without denying or impeding authorized access. Maintaining careful control over access to the system and its functionality improves our ability to recognize and prevent unauthorized activity.

Defining roles

Incorporating appropriate access controls is a multi-faceted endeavor that begins with defining the applicable roles against which user activity will be validated. A user's assigned role will often coincide with their privilege set, but it is also quite possible that a user's role will vary depending on other factors such as the specific activity at hand.

Consider an earlier example where a clinician also serves as a manager. When not working in a management capacity, the clinician's rules apply. Figure 11 demonstrates the role-based validation logic that must occur before a patient record is made visible.

1. While in the role of manager, all records are visible
2. If no longer active, no records are visible
3. If assigned as the primary clinician, their patient's record must be visible
4. If not the primary clinician of record, and not previously assigned to the case, the record is not visible
5. If previously assigned to the case, the record must be visible

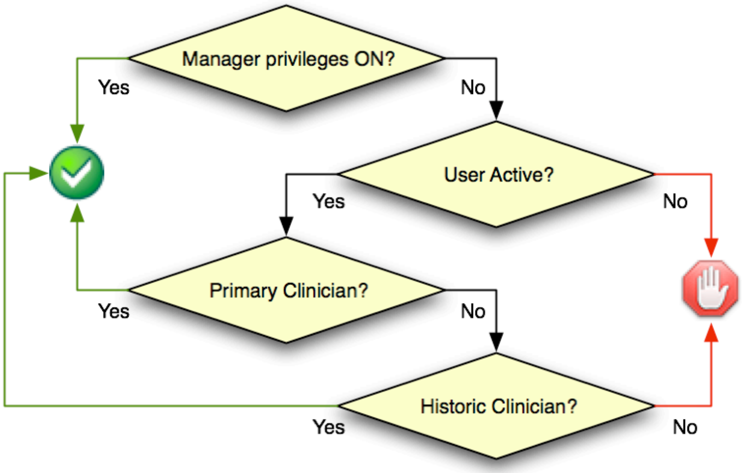
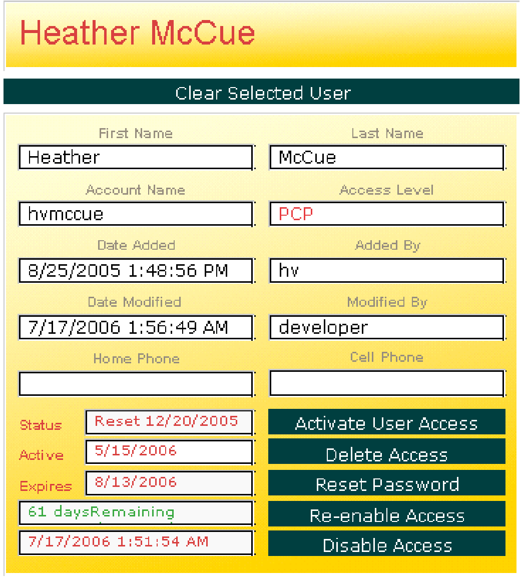


Figure 11—Role-based validation

A user's role may change from one day to the next, or as this example illustrates, from one record to the next. You must define your privileges accordingly. In your efforts to protect ePHI from unauthorized activity, remember that HIPAA also requires you to ensure availability of ePHI to authorized individuals. In this example, a prior clinician is authorized to view the patient record. Not shown in this example are the more granular validation rules which permit visibility of some but not all data within a patient's record to authorized users. For example, while a clinician can view a former patient's demographic ePHI, they cannot view clinical notes created by another clinician; though a primary clinician may view the clinical notes created by prior clinicians, until a new primary clinician is assigned and they too are considered historic providers. Regardless of a user's role as primary or historic provider, only the original creator of a clinical note can ever edit that note. Furthermore, a clinical note cannot be modified—even by its author—following submission of the clinical visit for billing.

The complexity required to implement role-based access will depend in large part on the business rules established by the covered entity.

It is likely that when validating access privileges, you will need to consider more than just the user's FileMaker privilege set. As appropriate, you must also validate the user's role relative to each specific record and data point, the functionality being requested, and possibly other factors which could impact the user's authority to view data or perform actions. Validating a combination of criteria often requires a careful blend of privilege set security and programatic controls.



Heather McCue

First Name: Heather	Last Name: McCue
Account Name: hvmccue	Access Level: PCP
Date Added: 8/25/2005 1:48:56 PM	Added By: hv
Date Modified: 7/17/2006 1:56:49 AM	Modified By: developer
Home Phone:	Cell Phone:

Status: Reset 12/20/2005
Active: 5/15/2006
Expires: 8/13/2006
61 daysRemaining
7/17/2006 1:51:54 AM

Figure 12—System User Detail and Controls

Assigning roles and granting access

FileMaker security offers several options for granting and managing access, in the form of external authentication, extended privileges, and scripted account management tools. While assigning user roles and granting appropriate access has never been easier, the requirements of a secure solution remain challenging. Developers are well advised to increase their familiarity with the FileMaker security tools already mentioned, plus FileMaker Server management, record-level access controls, custom privileges, and custom menus. You may also wish to follow up with some of the reading recommendations provided at the end of this document.

In order to provide the necessary activity logs and audit trails, your system will need a users table. You may want to incorporate account management features as well. Even in a system where external authentication is used to grant access and manage users, accessibility requirements may dictate the inclusion of internal account controls.

If, for example, external authentication were temporarily unavailable due to the failure of a domain server, it may be necessary to provide temporary access to authorized users directly within FileMaker. Using scriptable account management features, you can provide management with this functionality without exposing a system to the risks generally associated with allowing [Full Access] privileges.

Emergency planning

To ensure the availability of ePHI in the event of an emergency, four separate emergency plans are required by HIPAA.

1. Contingency Plan
2. Data Backup Plan
3. Disaster Recovery Plan
4. Emergency Mode Operation Plan

For the purpose of this discussion, an emergency is an event with the potential to render protected data unavailable. This might include catastrophic events such as fire, flood, hurricanes, earthquakes, and other natural disasters. It also includes short-term events such as equipment failures, power failure, sabotage, theft, or other unexpected disruption. Imagine the true story of a client whose facilities were temporarily closed by authorities due to a flea infestation.

The extent to which you are involved in the development or implementation of these plans will depend on your relationship with the client and available resources. If your client has their own IT department, they probably have a broader set of emergency plans defined, in which yours is just one of many systems being protected. However, the existence of broader contingencies does not always ensure that the needs of a FileMaker system will be adequately addressed and you should discuss this with your client as appropriate.

Data Backup

Covered entities are required to maintain "retrievable exact copies" of their ePHI. If the client's IT department will be responsible for managing FileMaker Server, you may need to advise them on safe practices for closing files and stopping the service. You may need to set up backup schedules or assist with redeployment in a new environment if needed. Depending on the level of risk identified, you may want to be involved with offsite storage, colocation, or other measures to assist users in the event of an emergency. Note that HIPAA also requires backups prior to moving equipment.

Contingency Plan

In cases where minimal risk is identified, your involvement with this aspect of compliance may be limited to establishing a responsible backup plan. Or you may choose to offer additional services such as offsite storage and short-term alternate hosting as needed. If you are physically located in the same region as your client, you may find yourself unable to assist in some emergencies (e.g., hurricane or flood). You may want to consider making arrangements with a third party for emergency hosting. You may need to remind your client that offsite backups stored within the affected region may be unusable.

It is also possible that your client will not consider contingent hosting plans necessary. Who exactly might need access to their system during the course of an emergency? Consider an outpatient mental health facility, whose records are accessed primarily by clinicians in the course of treating their patients. In the event of a hurricane, both patients and clinicians will evacuate, the facility will close, and regularly scheduled treatment will not be provided. It may be deemed perfectly acceptable by the covered entity's business rules for a system to remain inaccessible for a number of days in the event of a major regional emergency. If also serving remote users beyond the immediate locale, however, a down-time of greater than two hours may be considered unacceptable. Regardless of your client's acceptable limits for inaccessibility, HIPAA requires emergency access procedures and you should be prepared to discuss and assist with this as needed.

Disaster Recovery Plan

Procedures must be in place to restore any loss of data. This may consist of reverting to a backup, or it may involve a more elaborate import-recovery process.

Emergency Mode Operation Plan

Protection of ePHI must continue, even while operating in emergency mode. If the business processes you have defined to protect ePHI during normal operation may become unavailable during an emergency, alternate protections must be implemented.

Note: An emergency rendering data unavailable need not be a wide-reaching catastrophe. What if the only user with access to a record becomes unavailable? When designing your system's security, take care to provide for alternate, emergency access requirements.

Training and User documentation

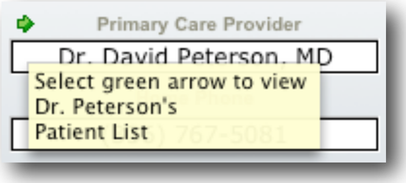
Part 11 requires system-specific user training and HIPAA requires implementation of a security awareness and training program for all members of the workforce. It would be appropriate for such a program to include training specific to the use of your FileMaker system, and where there is a need for training there will be a need for documentation.

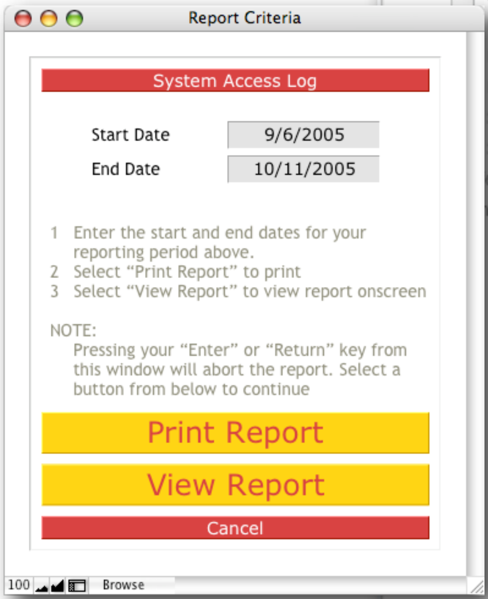
The level of documentation required to adequately train a user will vary considerably from one system to another and again, the extent of your involvement in this effort will depend on several factors. As stated earlier, this is an area where clarifying the boundaries of client-developer responsibility may be essential to the overall success of a project. It would seem a reasonable assumption on the part of a client that training and documentation should be included, while it is also reasonable for a developer to expect payment for their time—whether spent on development, writing, or training.

As a developer, you can help to reduce training requirements with a logical, easy-to-use interface design; consistent requirements, processes and design elements; and built-in tools.

1. Use window controls to create custom "dialogs". Break down multi-step processes into logical, individual tasks.

2. Use tab controls to reduce clutter and confusion while providing quick access to well-organized information.
3. Use Tooltips (available in FileMaker Pro Advanced) to provide users with training guidance or additional data.
4. Provide users with the instructional details they need, where they need it.
5. Label buttons and navigation options clearly.





Start Date 9/6/2005
End Date 10/11/2005

1 Enter the start and end dates for your reporting period above.
2 Select "Print Report" to print
3 Select "View Report" to view report onscreen

NOTE:
Pressing your "Enter" or "Return" key from this window will abort the report. Select a button from below to continue

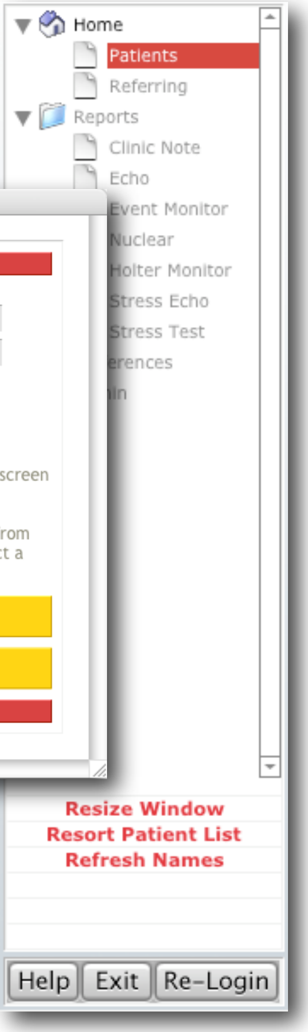


Figure 13—Examples of training & documentation "assists": layout-specific instructions, clearly labeled buttons, logical navigation, and Tooltips

Details from your system design and workflow notes may also prove useful when you are called upon to provide or assist with documentation. Figure 14 represents a system workflow illustrating how a 72-hour processing rule will be applied to Clinical Service Tickets (CST). In this system, a therapy session billing code is only valid if billed within 72 hours of service. A delay in processing can occur prior to submission or prior to processing. Regardless of where a delay occurs, the bill code is only valid for 72 hours. In this example, the same workflow used to confirm the proper sequence with a client during development can also be used to explain the rule to users.

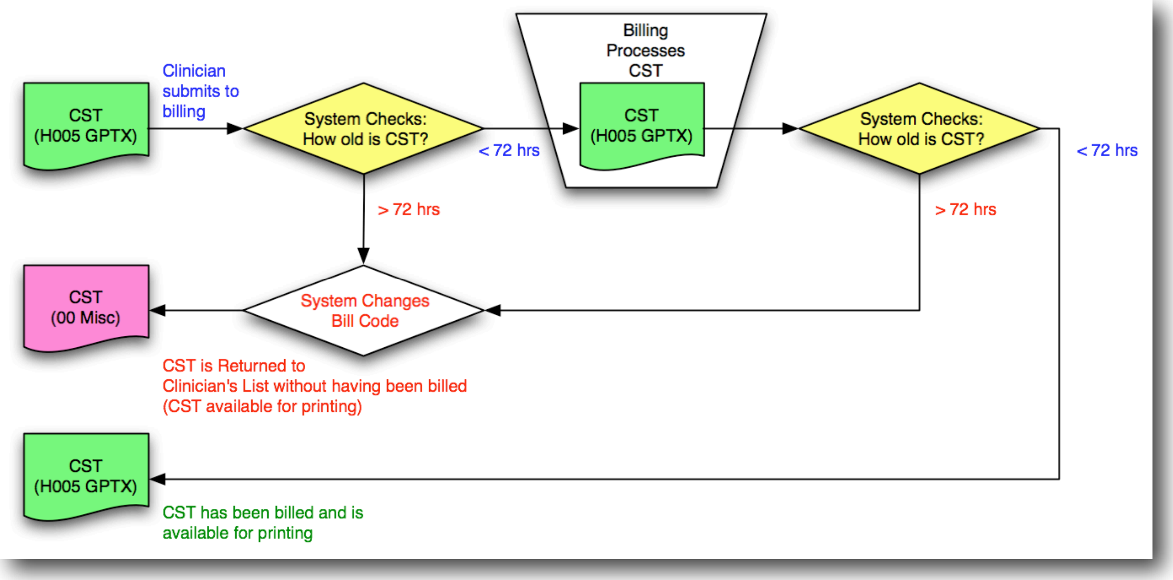


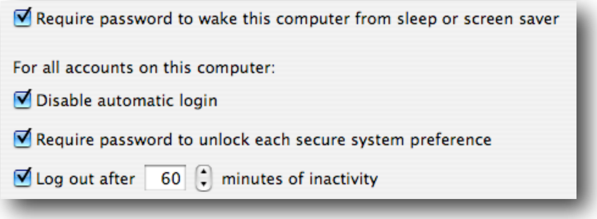
Figure 14—System workflow diagram illustrates a covered entity's "72-hour rule" for therapy billing codes

You might also consider building a help system into your solution. With the introduction of FileMaker Pro and FileMaker Pro Advanced 8.5, you now have the option to incorporate a dynamic web-based help system with the Web Viewer.

Auto log-off

The termination of inactive sessions is essential to any sound security schema and as an addressable technical specification in HIPAA, you should incorporate this within your solution. FileMaker Server allows you to meet this requirement with ease. Simply define a maximum idle time and FileMaker Server will automatically disconnect any user session that remains idle beyond your defined limit. If remote access via other services such as Terminal Services or Citrix are employed, you may also wish to define additional session limits such as hours of access, or maximum session length. However, it is not advisable to rely entirely on external systems for the enforcement of your idle session settings. A system deployed today via Citrix may in the future be accessed via Terminal Services, Remote Desktop, or something else entirely. It is also quite possible that a system deployed via Citrix to some users may be available to other users via Terminal Services. When using FileMaker Server it simply isn't necessary to rely on external controls to satisfy this requirement.

What if your system is intended for a single-user implementation, or only a handful of users considering peer-to-peer hosting? If sharing is expected—regardless of by how few—you should not consider anything less than FileMaker Server as a viable hosting option. Although FileMaker Pro and FileMaker Pro Advanced are capable of peer-to-peer sharing, the reliability of FileMaker Server is vital to maintaining the integrity and accessibility of your data.



☑ Require password to wake this computer from sleep or screen saver

For all accounts on this computer:

☑ Disable automatic login

☑ Require password to unlock each secure system preference

☑ Log out after 60 minutes of inactivity

Figure 15—Mac OS security settings example

If yours is a single-user system, and the use of FileMaker Server is deemed inappropriate, you might be wondering how you can enforce termination of idle sessions. There are a number of ways you might address this. As shown in Figure 15, you might choose to apply idle session settings at the Operating System level.

If you are a medical practitioner using a FileMaker Pro database of your own design for your own personal use, this would satisfy the regulatory intent (if the required system settings are also included in your documented policies and procedures). If, however, your system is ever expected to be seen by another user (or the machine upon which the system resides is not physically secured from potential access by another user), this approach will not suffice. Another approach would be to employ a third-party plug-in to check for inactivity. With a bit of creativity, other approaches can be devised, but none are as simple as defining the maximum idle time in FileMaker Server.

Data integrity tools

Ensuring that data has not been modified or deleted in an unauthorized manner requires a controlled environment where create, edit, and delete functionality is limited to authorized users under "proper" circumstances. This cannot be achieved by implementing a single control within FileMaker. A combination of access controls, user authentication, password management, audit trails, incident tracking, and user training are usually required. In addition to controls intended to limit system access to authorized users, you must also consider measures that will appropriately restrict authorized users from performing unauthorized actions.

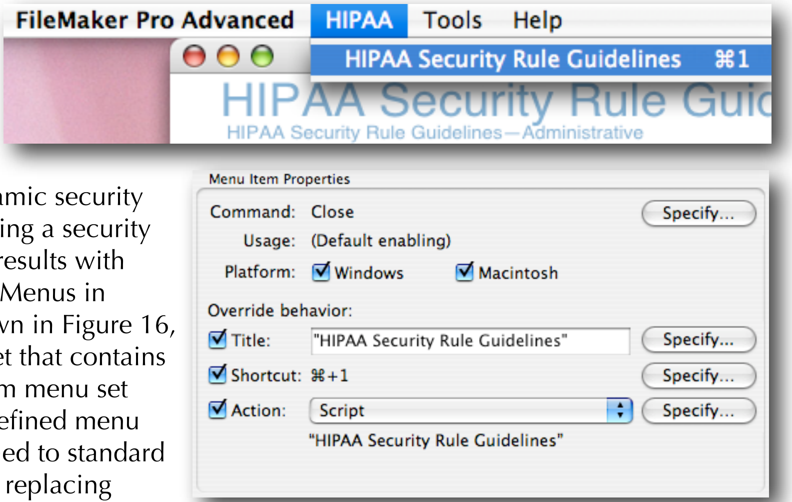
You might choose to employ a third-party plug-in such as SecureFM with MenuMagic[8] to control the user's environment by disabling specific menu items (and their keyboard shortcuts), hiding toolbars, disabling contextual menus, and disabling a user's access to potentially dangerous items like the Close window box.

Plug-ins often provide functionality not otherwise available, and they can further simplify your development efforts with centralized management of their feature set. For a variety of reasons, however, not every system requiring a secure environment is a viable candidate for plug-in use. Not all plug-ins are cross-platform or available in Universal binary versions (required for FileMaker Pro

[8] *SecureFM with MenuMagic is a product of New Millennium Communications, www.nmci.com*

and FileMaker Pro Advanced 8.5 running on Intel Mac computers), and plug-in licensing costs could also be a factor.

While a more complete and dynamic security environment may be achieved using a security plug-in, you can achieve similar results with relative ease by defining Custom Menus in FileMaker Pro Advanced. As shown in Figure 16, you can define a custom menu set that contains only a single function, or a custom menu set containing no items at all. Your defined menu item behaviors can also be attached to standard FileMaker commands. Effectively replacing undesirable standard features such as Close or Hide [window] with more desirable actions such as running a "Close" script of your own design with appropriate validations and session closing steps being executed prior to exiting.


FileMaker Pro Advanced
HIPAA
Tools
Help
HIPAA Security Rule Guidelines ⌘1
HIPAA Security Rule Guid
HIPAA Security Rule Guidelines—Administrative

Menu Item Properties

Command: Close [Specify...]

Usage: (Default enabling)

Platform: ☑ Windows ☑ Macintosh

Override behavior:

☑ Title: "HIPAA Security Rule Guidelines" [Specify...]

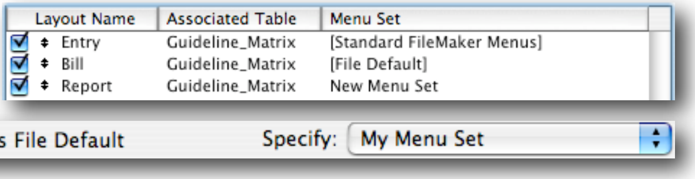
☑ Shortcut: ⌘+1 [Specify...]

☑ Action: Script [Specify...]

"HIPAA Security Rule Guidelines"

Figure 16—Example of a Custom Menu and menu item properties

Different menu sets can be assigned to specific layouts as appropriate and the Install Menu Set script step allows you to redefine a file's default menu set on a per-user basis.



	Layout Name	Associated Table	Menu Set
☑	Entry	Guideline_Matrix	[Standard FileMaker Menus]
☑	Bill	Guideline_Matrix	[File Default]
☑	Report	Guideline_Matrix	New Menu Set

☑ Use As File Default Specify: My Menu Set

Figure 17—Layout-specific menu settings and Install Menu Set's File Default option

In addition to their value as environmental controls, custom menus can also be used to improve
the user experience. For example, quick navigation in a complex system can be provided via familiar and easy to locate, menu-based controls. Consider Figure 19, comparing a standard FileMaker Window menu on the left, to a custom Window menu on the right.

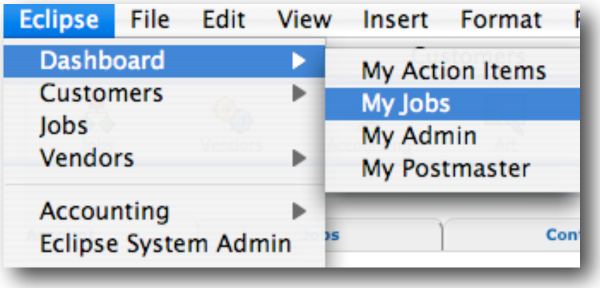


Figure 18—Sub-menus created via Custom Menus

The standard menu includes functionality we don't want to see in a controlled environment, and the user is provided with a single shortcut. In contrast, our custom menu includes fewer items (only those we want the user to have), shortcuts are available for each function, a "Rename Window" function has been added, and all functions are attached to scripts rather than the original default FileMaker actions.

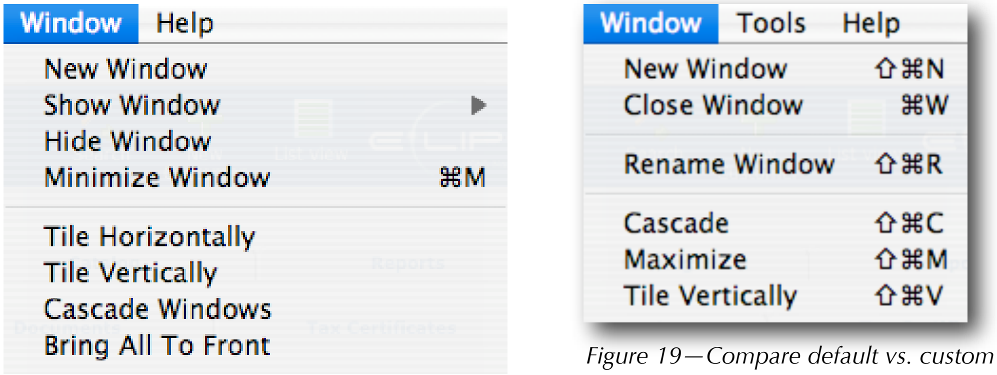


Figure 19—Compare default vs. custom menu

Data authentication

Following audit trail activity, you should be able to authenticate the validity of ePHI in your system. If transmitting ePHI via otherwise unsecured means, e.g., email, you should be able to authenticate that the data has not been modified in transit. You can accomplish this by contracting with a secure email encryption service to handle your communications, or you can encrypt using encryption product features such as CheckSum to verify that the data remains unaltered upon receipt. Unless your FileMaker system will be used to send mail containing ePHI, your use of an audit trail should be sufficient to meet data authentication requirements. If you expect to send email or fax (also an electronic transmission) from your system, a careful evaluation should be performed to define the rules of what, when, how, and to whom such communications will or will not be sent or received by your system.

User authentication

Your ability to accurately identify each specific user is the foundation upon which all other security controls are dependent, so it is critical that you follow sound practices when implementing user authentication. The task of authenticating a user—verifying that the user or entity is who they claim to be—may seem simple enough to many, but before you commit to a specific methodology consider the following.

1. When deciding how you will authenticate users, you must also determine when and why you will authenticate a user.
   a. Upon entry to the system (most definitely)
   b. Before viewing a different record (quite likely)
   c. Before performing certain actions (yes)
   d. Under other circumstances, e.g., end of current session's maximum time (possibly)
2. If a user is authenticated upon entry to the system, what subsequent events or actions could occur that might invalidate the authentication?
   a. elapsed time
   b. re-login as a different user, e.g., authentication for a manager's approval
3. Under what circumstances might you need to challenge an authenticated user's identity?
   a. elapsed time
   b. new function being performed
   c. different record being viewed
4. What business rules will be applied to the authentication, e.g.,
   password parameters (minimum length, alpha-numeric),
   duration, etc.
5. Is FileMaker Server hosting the file(s)?
6. Is a domain controller in use?
7. Will other, non-FileMaker factors influence your authentication choices?

Considerations such as client preference may weigh heavily in your choices. If your client uses single-source log-on for other protected applications, they will expect the same from their FileMaker system. Likewise, if multi-factor authentication is the norm, you will be expected to provide the same level of control. (e.g., smart card, biometric, or other security device).

If your client is already security conscious, it is unlikely that anything short of external authentication via FileMaker Server will be acceptable. If a domain controller is already in use, you will want to take advantage of this existing resource by implementing FileMaker Server external authentication. If FileMaker Server will be hosting the system, but there is no domain controller present, the FileMaker Server machine can be set up to manage user accounts for external authentication.

For simplicity and security, Server External Authentication of FileMaker is the preferred approach. However, external authentication is not an option for systems hosted peer-to-peer or single-user systems that are not hosted. In these instances users will instead log in directly to the FileMaker file.

While user authentication via FileMaker accounts alone may not be the preferred standard, external authentication is not a specific requirement of HIPAA and account-based authentication may be deemed appropriate and reasonable for some systems.

It was not uncommon in the past to find systems designed with authentication models that inadequately utilized FileMaker accounts in their attempt to simplify user management. Users would enter via a single default account, after which they were authenticated by an interface-driven log-in. This model authenticated each login against a user table, requiring passwords to be stored as data. You may choose to employ a user table for other purposes, including flexible management of certain role- or function-based privileges, but the ease with which FileMaker accounts can now be managed renders this inferior approach unnecessary as a sole methodology for user authentication.

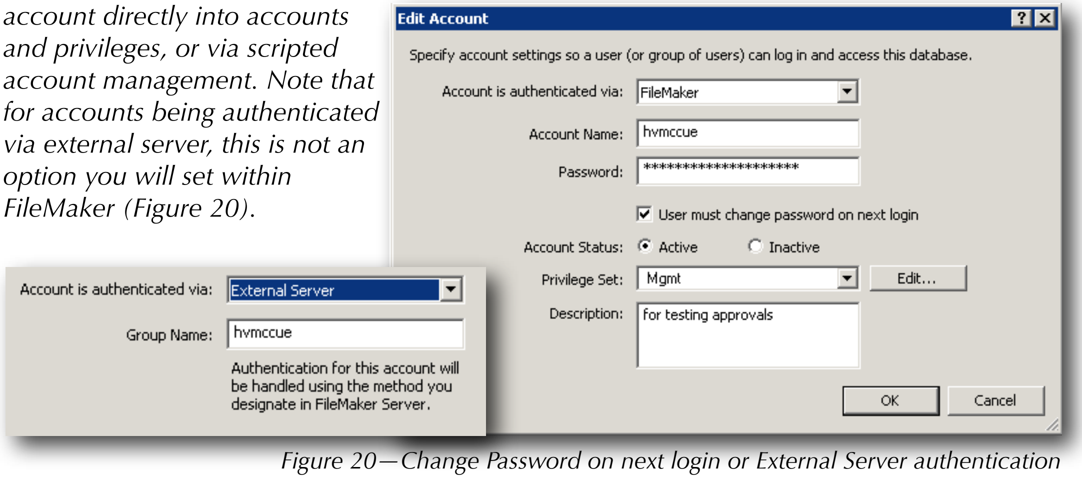
Regardless of a system's deployment model, it is both reasonable and appropriate to employ a sound user authentication model based on FileMaker accounts and privilege sets.

Password management

HIPAA's administrative guidelines require implementation of procedures for creating, changing, and safeguarding passwords. Similar provisions can be found in Part 11. Given the importance of uniquely identifying each specific user, the significance of managing user passwords should be obvious. When user passwords are easily compromised, the reliability of user authentication is at risk and your ability to protect ePHI is jeopardized.

Take care in designing your model for password management, taking into account the following issues:

1. Integrity
   To ensure that only the individual user knows their own password, new accounts can be created with the requirement that the password be changed at the next login. This can be specified whether adding an account directly into accounts and privileges, or via scripted account management. Note that for accounts being authenticated via external server, this is not an option you will set within FileMaker (Figure 20).



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: hvmccue

Password: ********************

☑ User must change password on next login

Account Status: ● Active ○ Inactive

Privilege Set: Mgmt

Description: for testing approvals

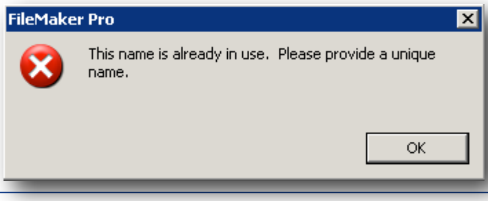
Account is authenticated via: External Server

Group Name: hvmccue

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

Figure 20—Change Password on next login or External Server authentication

2. Uniqueness
   You must be able to uniquely identify each user, so shared login accounts are obviously prohibited. If not utilizing external authentication, you must provide each individual user with a separate FileMaker account. Part 11 requires a unique combination of user name and password. Because you cannot prevent multiple users from choosing the same password, you must ensure that duplicate user names are not allowed. At first glance this may not appear to be an issue, particularly since FileMaker itself does not allow duplicate account names (Figure 21). However, it is possible for duplicate account names to exist when a combination of internal and external accounts are permitted.

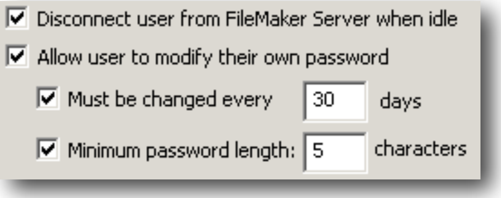


FileMaker Pro

This name is already in use. Please provide a unique name.

Figure 21—Duplicate FileMaker Account name

3. Balance
   Protecting the integrity of user passwords also requires a careful balance of minimum password requirements and realistic expectations. If your password management requirements are too excessive for users to follow, they are likely to resort to unsafe practices.

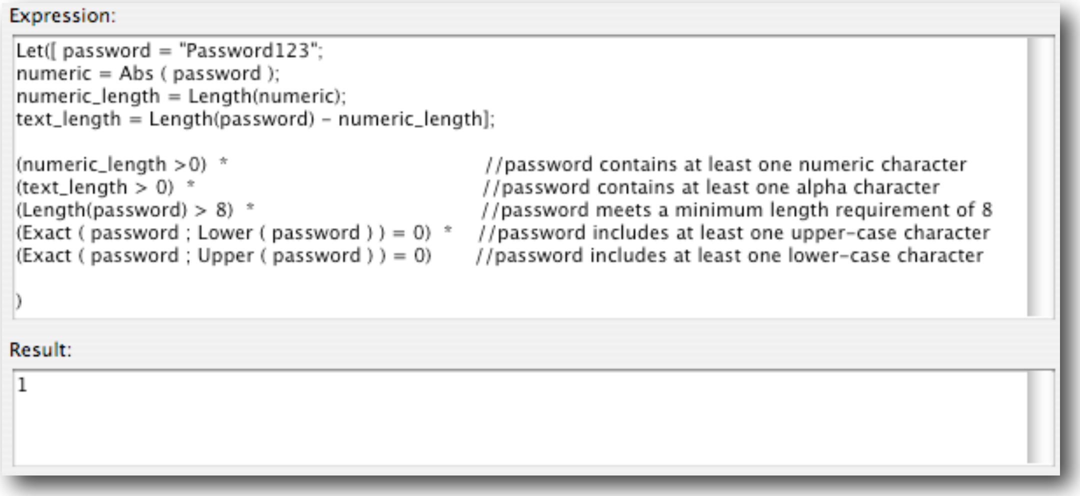


☑ Disconnect user from FileMaker Server when idle
☑ Allow user to modify their own password
☑ Must be changed every 30 days
☑ Minimum password length: 5 characters

Figure 22—Privilege Set password requirements

4. Business Rules
   When corporate standards exist for other password-protected systems already in use, your system will likely be required to follow those same standards. If external authentication is not used, you will need to enforce those rules directly within FileMaker. You can specify a minimum password length, and change frequency within each FileMaker privilege set, but this may not suffice to satisfy client-driven business rules. How would you enforce a maximum password length? What if both alpha and numeric characters are required? What if at least one non-alpha-numeric or upper-case character is required?

   If any requirements other than minimum length are imposed, you will need to validate a user's password selection before creating or updating an account. To test for the specific requirements, you will need to write an appropriate calculation that takes into account all applicable rules. The variations are endless, but Figure 23 provides an example of how you might validate an alpha-numeric password of at least 8 characters containing both upper- and lower-case characters.



Expression:

```
Let([ password = "Password123";
numeric = Abs ( password );
numeric_length = Length(numeric);
text_length = Length(password) - numeric_length];

(numeric_length >0)  *                              //password contains at least one numeric character
(text_length > 0)  *                                //password contains at least one alpha character
(Length(password) > 8)  *                           //password meets a minimum length requirement of 8
(Exact ( password ; Lower ( password ) ) = 0)  *    //password includes at least one upper-case character
(Exact ( password ; Upper ( password ) ) = 0)       //password includes at least one lower-case character

)
```

Result:

```
1
```

Figure 23—Validating password requirements

Electronic Signatures

Electronic signatures are not required by HIPAA, but they are the primary subject of Part 11. As stated earlier, electronic signatures are considered the legally binding equivalent of handwritten signatures. As such, a number of controls have been defined to ensure their irrefutability. So what is an electronic signature? The following definitions are quoted from the regulatory text of Part 11:

"Digital signature means an electronic signature based upon cryptographic methods of originator authentication, computed by using a set of rules and a set of parameters such that the identity of the signer and the integrity of the data can be verified."

"Electronic Signature means a computer data compilation of any symbol or series of symbols executed, adopted, or authorized by an individual to be the legally binding equivalent of the individual's handwritten signature."

Though not required for HIPAA compliance, should you choose to employ electronic signatures within a HIPAA-compliant system, an electronic signature standard must be followed. HIPAA does not define such a standard within its final rule, but the need to adopt such a standard was addressed within the comments of the proposed rule where the electronic signature was defined as follows.

In the electronic environment, the same legal weight associated with an original signature on a paper document may be needed for electronic data. Use of an electronic signature refers to the act of attaching a signature by electronic means. The electronic signature process involves authentication of the signer's identity, a signature process according to system design and software instructions, binding of the signature to the document and non-alterability after the signature has been affixed to the document. The generation of electronic signatures requires the successful identification and authentication of the signer at the time of the signature.

There are a number of features required in an electronic signature. Among these are message integrity, nonrepudiation, and user authentication. Additional implementation features that may be used include the ability to add attributes, continuity of signature capability, countersignatures capability, independent verifiability, interoperability, multiple signatures, and transportability.

What does this mean to you in the development of a FileMaker system?

In short, it means that if you intend to have users "sign" electronic records, then you must first be able to ensure a user's authenticity to the extent that they cannot later dispute their identity as the signing party. You must also ensure that the signature itself (and any accompanying message) cannot be altered in any way—again, such that the validity of an electronic signature is irrefutable.

When might you use an electronic signature?

You may employ an electronic signature whenever the business rules dictate. For example, you may use an electronic signature to allow the user to "sign" a clinical note, or to record a supervisor's authorization to transfer a patient record from one practitioner to another. Unlike an audit trail record which transparently identifies who performed an action or modified data, implementing an electronic signature should require an explicit action on the part of the user. This may be as simple as requiring an already authenticated user to "sign" a document via button-click confirmation, or as complex as requiring the user to re-authenticate.

How do you represent an electronic signature?

Though technically digital in nature, an electronic signature needs to be displayed in human-readable form. Where appropriate, additional descriptive text may also be required. For example, if approving a treatment plan, the signature should state "Approved" along with the user's name, date/timestamp of signature and any other appropriate details. A graphic representation of the user's handwritten signature may also be included, but this is not necessary and this alone does not constitute an electronic signature. If graphic signatures are to be used, explicit rules for their use should be defined within the policies and procedures. When an alternate signatory is permitted, e.g., a supervisor signs off on a treatment plan, the electronic signature must reflect the identity of the actual signer.

External Security

Beyond the measures you will implement within FileMaker to protect ePHI, you should also contemplate the environment in which your FileMaker system will be deployed. Take steps to mitigate identified risks as appropriate. For example, you might recommend the installation of a secure server rack, or assist with the replacement of an inferior firewall. It is unlikely that you will be called upon to write policies or address the physical security requirements of HIPAA, but you should be prepared to assess network or other issues as needed.

Transaction Codes, Code Sets and Identifiers

Though not directly related to the Security Rule, you may find yourself needing to incorporate transaction code standards. This is not a technical requirement, but it represents a data requirement and is therefore important to the discussion. This requirement does not apply to all systems, but it may apply to yours.

What are transaction codes?

When healthcare information is transferred electronically (e.g., from a provider to an insurance company or to Medicaid), HIPAA requires providers to uniformly identify those transactions using the standards identified for Electronic Data Interchange (EDI). Ten standard transactions have been identified by HIPAA for these transmissions, including claims and encounter information, payment and remittance advice, and claims status among others.

What are code sets?

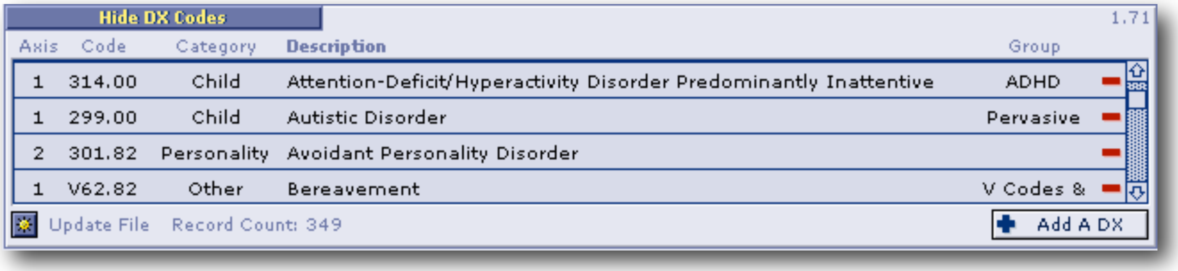
Code sets used to identify diagnosis and clinical procedures must also follow HIPAA-specified standards such as ICD-9[9], CPT-4[10], and DSM-IV[11].

What are HIPAA identifiers?

Not to be confused with "individually identifiable data" which defines ePHI, HIPAA Identifiers are standards created to provide unique identifiers for all health care providers[12], employers[13], and health plans[14].

How codes, code sets, and identifiers may affect your FileMaker system

Which transaction codes, code sets, and identifiers are required for your system (if any) will be dependent upon the transaction types managed by the system, and whether the system's data is or is not transferred electronically to other entities.



Axis	Code	Category	Description	Group
1	314.00	Child	Attention-Deficit/Hyperactivity Disorder Predominantly Inattentive	ADHD
1	299.00	Child	Autistic Disorder	Pervasive
2	301.82	Personality	Avoidant Personality Disorder	
1	V62.82	Other	Bereavement	V Codes &

Record Count: 349

Figure 24—Example from DSM-IV Code Set (DX = diagnosis)

In the event that a particular code set is required, you will need to include the proper code in addition to any descriptive details that a user may need to see. For example, a diagnosis of "avoidant personality disorder" must be identified with the DSM-IV code 301.82, while a diagnosis of "attention deficit/hyperactivity disorder predominantly inattentive" would be coded as 314.00.

To prevent code errors that could render your system non-compliant, and to facilitate updates as the standards change (e.g., when new diagnosis codes are added), you should consider adding a codes table to your system. This can be as simple as a two-field table containing nothing more than the code and a description for use in created a value list, or you may choose to include additional descriptive details, categories, and other data relevant to your implementation.

Although there are hundreds of codes within the mandated standards, you need only be concerned with the codes that apply to your system. In the example of a mental health practice, if the practice treats only children, then only the DSM-IV codes that can be applied to children need to be included. If a smaller subset of these represent the most frequently used diagnoses, then you may opt to limit a diagnosis value list to these. If your system documents only the Axis II diagnosis, then you needn't incorporate Axis I or Axes III through V.

Which standards are to be used and any rules surrounding how they will be obtained or updated should be documented within the covered entity's policies and procedures.

[9] *International Classification of Diseases, 9th revision, Clinical Modification (ICD-9).*

[10] *Current Procedural Terminology, American Medical Association (CPT-4).*

[11] *Diagnostic and Statistical Manual of Mental Disorders, Fourth Edition (DSM-IV), uses a multidimensional approach to diagnosis based on the assessment of five dimensions, known as Axis I through Axis V.*

[12] *Standard Unique Health Care Provider Identifier (CMS-0045F) adopts the NPI (National Provider Identifier) as the standard for all health care providers under HIPAA; compliance date is May 23, 2007.*

[13] *Standard Unique Identifier for Employers (CMS-0047F) specifies that an employer's tax ID or Employer Identification Number (EIN) will be used.*

[14] *Standard Unique Health Plan Identifier (CMS-4145P) establishes a standard health plan identifier for payers.*

Encryption

HIPAA is all about protecting ePHI, from the administrative policies and procedures defined to manage security measures, to the control and auditing of access and data modifications. The Security Rule also provides for data-level protection by requiring implementation of a mechanism to encrypt and decrypt ePHI.

HIPAA defines encryption as "the use of an algorithmic process to transform data into a form in which there is a low probability of assigning meaning without use of a confidential process or key." You needn't develop and program your own algorithm to implement encryption. It is much simpler, and more reliable, to utilize a plug-in such as Troi Encryptor[15] (which offers several algorithms, including AES[16], Safe Ascii, Checksum, and Text Signature).

What must be encrypted

Encryption must be applied to ePHI, aka individually identifiable information, but what constitutes "individually identifiable"? Data is considered identifiable if it includes any of the 18 patient identifiers defined by the U.S. Department of Health and Human Services. The intent is to prevent identification of a specific individual, so the following identifiers are protected when they apply to the individual, their employer, a family member, or if the provider is aware that the information could be used, either alone or in combination with other information, to identify a specific individual.

Practically speaking, it is reasonable to assume that the identity of a child could be determined from parental data. It is less reasonable, however, to expect that a specific child's identity could be determined from the presence of a teacher's name. Of course, if a child's hair color and gender were also unprotected, it is quite conceivable that someone could identify the red-headed girl in Mrs. Smith's first grade. There should be no question about encrypting the first 17 patient identifiers, but you should clarify with your client which of their data is reasonably covered by the 18th identifier.

Patient Identifiers

1. Names
2. All geographic subdivisions smaller than a state, including street address, city, county, precinct , ZIP Code, and their equivalent geocodes, except for three digits of a ZIP Code if, according to current publicly available data from the Bureau of the Census,
   a. the geographic unit formed by combining all ZIP Codes with the same three initial digits contains more than 20,000 people;
   b. the initial three digits of a ZIP Code for all such geographic units containing 20,000 or fewer people are changed to 000;
3. All elements of dates (except year) for dates directly related to an individual, including birth date, admission date, discharge date, date of death; and all ages over 89 and all elements of dates (including year) indicative of such age, except that such ages and elements may be aggregated into a single category of age 90 or older;
4. Telephone numbers
5. Fax numbers
6. Electronic mail addresses
7. Social security numbers
8. Medical record numbers
9. Health plan beneficiary numbers
10. Account numbers
11. Certificate/license numbers
12. Vehicle identifiers and serial numbers, including license plate numbers;
13. Device identifiers and serial numbers;

[15] *Troi Encryptor is a product of Troi Automatisering, www.troi.com*

[16] *Advanced Encryption Standard (AES)*

14. Web Universal Resource Locators (URLs);
15. Internet Protocol (IP) numbers;
16. Biometric identifiers, including finger and voice prints;
17. Full face photographic images and comparable images; and
18. Any other unique identifying number, characteristic, or code, unless no individual could be identified in any manner and the number or code is not derived from or related to information about the individual.

The encryption-decryption process

The basic requirements of encryption will become quickly evident as you begin to explore and test vendor examples. However, failure to prepare for the unique requirements of actually working with encrypted data could leave you unable to retrieve your data. So plan carefully before implementing your own encryption schema.

Encrypting data safely is best accomplished with a minimum of two to three fields for each encryptable data point. To address the unique requirements of your implementation you may prefer to use four fields. Consider the following:

1. A user must be able to enter the original data in a human-readable form

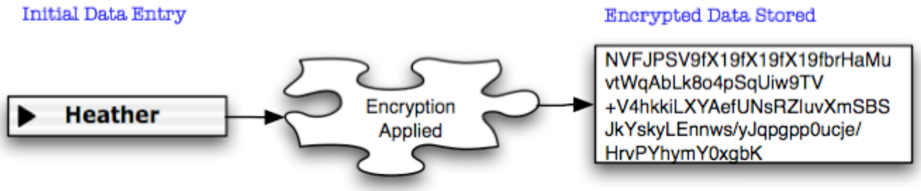


Figure 25—Basic encryption sequence

2. That entry must then be stored in an encrypted form
3. The original entry cannot be stored

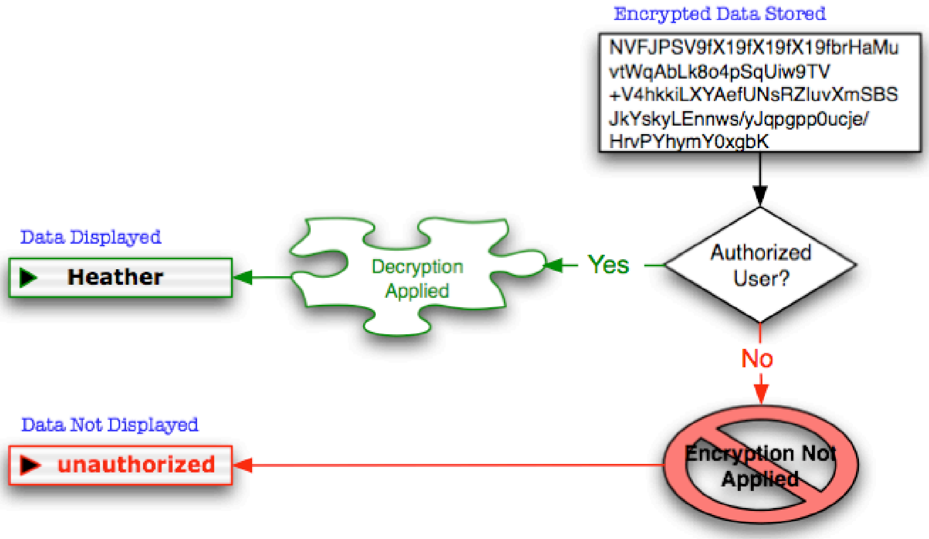
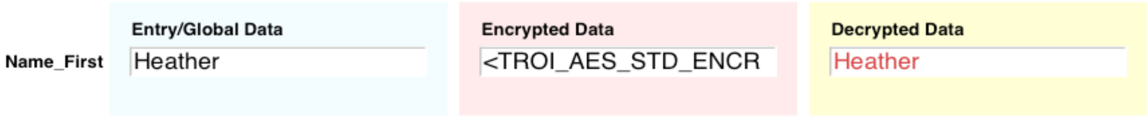


Figure 26—Basic decryption sequence

4. Once encrypted, only authorized users can view the data in its decrypted form
5. When edited, the new data must also be encrypted, replacing the original encryption



	Entry/Global Data	Encrypted Data	Decrypted Data
Name_First	Heather	<TROI_AES_STD_ENCR	Heather

Figure 27—Encryption fields; 1) global entry/edit; 2) encrypted, stored; 3) decrypted, unstored, display only

You can use a global field for initial data entry and editing, but you will not want to rely on a global for the display of decrypted data because this will prevent you from being able to view a list of records.

Encryption keys

Data is encrypted using a “key” and only that key can be used to successfully decrypt the encryption. Failure to adequately protect the encryption key could result in unauthorized access to encrypted data. The actual key should never be revealed to a user and the proper key should only become accessible to a decryption process following confirmation that a user is authorized.

1. Do not store encryption keys as data
2. Avoid storing encryption keys in a susceptible text format within scripts or calculations
3. Use custom functions to store encryption keys
4. Consider alternatives for further protection of your keys
    a. encrypt the keys
    b. if using variables to pass your keys, remember that users with FileMaker Pro Advanced could use their Data Viewer to identify them

User-defined keys

Data must be retrievable, so allowing users to define their own encryption keys is not appropriate for a HIPAA-compliant system. Should a user become unavailable, whether temporary or permanent, their encrypted data would become irretrievable.

Changing keys

A periodic change of encryption keys may sound like a great idea, but if a key is changed, all data encrypted with the former key would need to be decrypted and re-encrypted using the new key. This should only be attempted if a well-defined and tested process for such a transition is in place. Consider also the issue of recovering data from backups. If the keys are changed, will data retrieval be negatively impacted?

Encryption Issues

A primary issue you must address when working with encrypted data is how you will search the encrypted fields.

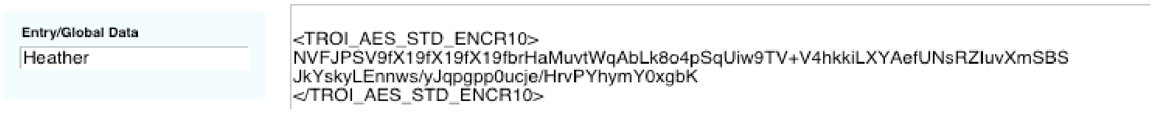
Searching encrypted data

Remembering that your decrypted display fields cannot be stored, you will need to use your encrypted fields for searches and relationships. Provide users with a global field for entry of their search criteria. Encrypt the criteria, then use this encrypted result as the basis for your search or relationship filter.

You may not need to search against all encrypted fields, but for any field where this functionality is required, you should choose encryption algorithms that provide consistent results. The AES, highly recommended for its security, will cause queries and relationships to fail because it returns a different encryption result each time. Troi Encryptor allows you to disable this feature.

You will have no control over the characters returned in the encryption results and some of these can cause FileMaker searches and relationships to fail. Consider the example shown in Figure 28, where the core encryption result has been divided into two lines by a “silent” character. This will not prevent a search from locating the desired record, but it may cause the found set to include records that do not meet the specified criteria. The larger the record count, the more likely this is to pose a problem. To resolve the problem, apply a Substitute function to both the stored encryption and the criteria encryption. Your substitute will probably be nested to address multiple such “illegal” characters.

Encrypted numbers (and dates) can pose similar problems, but we have found that converting numbers to text eliminates potential conflicts and yields reliable results.



Heather

```
<TROI_AES_STD_ENCR10>
NVFJPSV9fX19fX19fX19fbrHaMuvtWqAbLk8o4pSqUiw9TV+V4hkkiLXYAefUNsRZIuvXmSBS
JkYskyLEnnws/yJqpgpp0ucje/HrvPYhymY0xgbK
</TROI_AES_STD_ENCR10>
```

Figure 28—Encrypted results may include problematic line breaks

Sorting

Sorting records by encrypted fields poses another challenge, for obvious reasons. If your record count is low, sorting by the unstored decrypted values won't be much of an issue, but a cardiology practice with over 2,400 patient records would find the performance unacceptable. To resolve this, records can be periodically resorted, with each then being assigned a numeric sort value.

"Bad" Characters

If uncooperative characters are identified, e.g., silent line breaks, you may find that they cannot be incorporated directly within a Substitute function. When this occurs, store the actual character in a global field (probably in your system preferences table). Define a custom function equal to that field, then use the custom function within your substitute calculations.

Editing Encrypted Data

To edit encrypted data, you must first provide the user with the decrypted data, probably in a global field. After the data has been edited, you will replace the previous encryption with a new encryption of the new data. This can be scripted, or it can be accomplished using auto-enter calculations for both the encryption and decryption fields. This provides a more dynamic and efficient result, but will require more attention to detail. You must decide, for example, how you wish to control the sequence of evaluation so that you are not replacing your new data with old data. If using global fields for data entry, you will also need to provide a mechanism for identifying the specific record being modified.

If you need to import ePHI, you will need to include a non-global field to "hold" the new data until it has been encrypted. Use a looping script to sequence through the affected records, processing each by transferring the imported data to your global entry field and triggering a new encryption. Once encrypted, the holding field can be cleared and you can move to the next record.

Your requirements will vary and so too will your approach.

FileMaker—A Tool of Compliance

Epilogue

> *The project was completed on time and within budget, for less than a quarter of what had been quoted by the "well-known enterprise system" folks, so we all had reason to be proud of our efforts that night. The FDA validation document was held high for all to see as toasts to the team began. The actual FileMaker team consisted of only two individuals, yet the private dining room held nearly 30. And the title of Most Valuable Team Member was not awarded to either of the programmers who built the Part 11-compliant system. It went instead to the technical writer, without whom validation would not have been achieved.*

Beyond the Programming

Can a FileMaker system meet the specific technical requirements of Part 11 and the Security Rule? Of course, but before you dive into development fun, you (and your client) must understand that achieving compliance requires more than programming talent. The extent to which you might fulfill additional compliance-related roles is between you and your client, but there is a natural connection between your efforts and the required policies, documentation, testing, and training.

"Compliant" software is only compliant within a compliant environment.

Policies

In addition to technical policies referenced within the guidelines already discussed, HIPAA also requires a covered entity to implement policies addressing the following issues: *Security Management Process, Sanction Policy, Workforce Security, Information Access Management, Security Incident Procedures, Contingency Plan, Workstation Use, Device and Media Controls, Disposal,* and *Facility Access Controls*.

Non-technical policies specified within the addressable guidelines include *Access Authorization, Access Establishment and Modification, Facility Security Plan,* and *Maintenance Records*.

As a developer, you are not responsible for writing, implementing, or enforcing these policies on behalf of the covered entity. As a consultant, however, familiarizing yourself with the relevant guidelines should prove helpful to you and the covered entities you support.

"Implement policies and procedures…" is a phrase repeated often in the guidelines. Where there is a policy element, there will be a specific or implied procedural element. There are also guidelines which require the implementation of procedures without a defining policy. For example, *Information System Activity Review* [164.308(a)(1)(ii)(D)] states:

> Implement procedures to regularly review records of information system activity, such as audit logs, access reports, and security incident tracking reports.

Though not liable for the development or maintenance of a covered entity's policies, you may well find yourself asked to assist with the process and developing features in support of the policies and procedures they choose to implement. Imagine the many features you could suggest incorporating in support of the example above. How many are already present within the existing solution? Which of your potential suggestions are reasonable and appropriate for the covered entity to consider? Is the budget sufficient? If not, what alternatives might you suggest? The possibilities—and opportunities—are numerous.

Documentation

There are no specific documentation requirements within HIPAA's technical guidelines, nor do the guidelines explicitly define the manner in which most policies and procedures should be communicated. It is logical to expect, however, that most policies and many procedures will be documented in writing. There is no HIPAA-mandated documentation-of-effort obligation imposed on you as a developer, but just as we are encouraged to retain receipts in support of tax-deductible expenses, you may want to consider documenting the technical features and controls developed in support of HIPAA compliance.

Budget and the nature of your developer-client relationship will ultimately determine the extent to which you will provide documentation of your efforts, but this is a topic you should be prepared to discuss early. Portions of your coding may be proprietary, but when a client is seeking your consult in support of compliance there may be an unspoken expectation that you will provide documentation for their use—not to reverse engineer your efforts, but to corroborate their compliance effort.

Documentation is not an unreasonable expectation on the part of a covered entity. As any developer knows, however, documenting a solution can become a time-consuming project all its own. Depending on the scope of your efforts (e.g., adding a handful of features to a client-built solution versus developing a ground-up custom solution), determine documentation requirements up front.

The mutually agreed upon scope of your documentation could be as simple as an outline or description of features (perhaps already included in your proposal). It may include defined business rules or a security privileges grid; in some cases it would even be appropriate or necessary to include schema definitions.

Note that while HIPAA regulations are vague with regard to documentation requirements, achieving Part 11 FDA validation will require significant documentation of intent, methodology, and schema. If a technical writer is provided, do not assume that this absolves you of documentation responsibility. The writer will likely depend on you to provide—at minimum—system training and schema definitions. You will want to discuss with your client the extent to which your documentation support is covered in the budget.

Testing

HIPAA does not require specific systems testing, however, there is an obvious expectation that compliance-oriented features and functionality do, in fact, work. Part 11 FDA validation, on the other hand, requires extensive systems testing. As the developer, your involvement with this effort may or may not be required. You should discuss this with your client prior to defining the scope of your work.

Training

Both Part 11 validation and HIPAA require user training. Again, the extent to which you are expected to assist with or participate in this effort should be discussed at the outset. If you are working with an enterprise-level facility such as a hospital or pharmaceutical company, the client will likely take advantage in-house training assets. When consulting with a small medical practice, you may be asked to provide minimal to extensive training. If developing a solution for resale, you are probably already providing training materials.

Conclusion

There is no question that it was far more complex to implement the controls necessary for compliance using FileMaker Pro 5.5 and 6. More of the essential security tools are now native to FileMaker, and development tools such as script parameters, custom functions, variables, and custom menus have reduced both the system overhead and mental effort required. In short, the development effort has been simplified by magnitudes. Still, FileMaker does not come with an "Apply HIPAA Controls" preference setting, nor is it reasonable to expect that it ever will. Achieving compliance will remain more than a trivial effort.

FileMaker provides a robust toolset, which an experienced developer can use to incorporate the necessary compliance-oriented technical mechanisms. However, achieving compliance is not an "Ages 5 and up" project. In the hands of a novice or hobby developer, a well-intentioned effort is likely to fall short. It is not necessary for a practice to invest in a costly, commercial EMR when an existing in-house solution already meets (or can be cost-effectively developed for) their needs, but with the confidentiality of patient data at risk, falling short is simply not a viable option. A practice with an existing, custom FileMaker solution should consider working with a professional FileMaker developer to incorporate the appropriate compliance measures.

The real challenge is the technical mindset of both the developer and client. Is FileMaker perceived as a Tool of Compliance?

FileMaker empowers a creative and conscientious developer to build all the security-based functionality of Part 11 compliance for a significant fraction of the cost and time. There are those, however, who find it difficult to accept the fact that FileMaker is as good as we say. Be confident—FileMaker is clearly up to the task and can hold its own against major enterprise-grade solutions.

Similarly, many of those challenged to comply with the HIPAA Security Rule cling to the crutch of "reasonable and appropriate" hoping for justification to do as little as possible. Be assertive—FileMaker makes everything reasonable and appropriate.

FIleMaker is a Tool of Compliance—believe it.

About the Author

A FileMaker user since 1988, Associate member of the FileMaker Solutions Alliance since 1995, past FileMaker Developers Conference speaker, HIPAA Workshop moderator, Certified FileMaker 7 Developer, and Certified FileMaker 8 Developer, Heather McCue is a well-respected contributor to the FileMaker developer community. Over the past 25 years she has parlayed a broad range of experience into an equally diverse array of FileMaker solutions, including FDA- and HIPAA-compliant systems for the pharmaceutical and health care industries.

A pioneer advocate of FileMaker as a Tool of Compliance, Heather continues to broaden her expertise as Senior Developer for Dallas-based Harmonic Data Associates—a FileMaker Solutions Alliance Partner.

Recommended Reading/Training

1. *FileMaker Security: The Book*
   Written by FileMaker Pro security expert Steven H. Blackwell and published by New Millennium Communications http://www.nmci.com
2. *Special Edition Using FileMaker 8*
   Written by Scott Love, Steve Lane, and Bob Bowers and published by Que
3. FileMaker Professional Training Foundation Series III for FileMaker 8
   http://www.filemaker.com/developers/professionaltraining/index.html

Links

1. Security Standards; Final Rule. 45 CFR Parts 160, 162, and 164. February 20, 2003. Please note that Part 164 begins on page 8374 of the Federal Register, after a 40-page preamble.
   http://www.cms.hhs.gov/SecurityStandard/Downloads/securityfinalrule.pdf
2. Security Rule Guideline Matrix. Easy-to-read chart presents administrative, physical, and technical guidelines of the Security Rule grouped by "required" or "addressable". In addition, identifies wether or not responsibility is shared between the covered entity and developer. Matrix covers 45 CFR §164.308 through §164.312.
   http://www.harmonic-data.com/hipaamatrix.pdf
3. Proposed Rule. On August 12, 1998, the U.S. Department of Health and Human Services published the proposed Security Rule. Based on public comment, the Proposed Rule was stripped of specificity and made vague.
   http://www.cms.hhs.gov/SecurityStandard/Downloads/securityproposedrule.pdf
4. HIPAA Glossary. The Workgroup for Electronic Data Interchange (WEDI), a national consortium of public and private corporations and organizations within the health care industry, focus on the electronic delivery of electronic health care information. WEDI maintains a comprehensive glossary of HIPAA-related terms and acronyms.
   http://wedi.org/snip/public/articles/hipaa_glossary.pdf
5. 21 CFR, Part 11—The Final Rule. On March 20, 1997, the Federal Drug Administration (FDA) published regulations that provide criteria for acceptance by the FDA of electronic records, electronic signatures, and handwritten signatures executed to electronic records as equivalent to paper records and handwritten signatures written on paper. Please note that the actual regulation begins on page 13464 of the Federal Register, after a 34-page preamble.
   http://www.fda.gov/ora/compliance_ref/part11/FRs/background/pt11finr.pdf
6. 21 CFR, Part 11—Industry Guidance. This document is intended to describe the Food and Drug Administration's current thinking (circa 2003) regarding the scope and application of CFR 21 Part 11; Electronic Records; Electronic Signatures.
   http://www.fda.gov/Cder/guidance/5667fnl.pdf