

In-Depth Container Fields



In Depth: Container Fields

About this Technical Brief	3
Understanding Document Management (DM) and Digital Asset Management (DAM)	3
Benefits of DM/DAM Systems.....	3
Glossary	3
Understanding Container Fields With Managed Storage	5
Base Directories on Local Files	5
Managing Base Directories for Local Files	6
Adding and Editing Base Directories	7
Specifying Base Directories for Managed Container Fields.....	8
Base Directories on Hosted Files	9
Understanding Managed Container Storage Types	10
Secure Storage.....	10
Using Open Storage.....	11
Transferring Externally Stored Files	12
Handling Large Files	15
Thumbnails	15
Streaming Data.....	16
Optimizing Container Fields	16
Distributing Server Requests.....	19
Understanding Data Cache	20
Saving Volume Space	20
Moving Databases with Managed Storage	20
Backups and Managed Containers	21
Understanding Hard Links.....	22
Working With Enhanced Container Fields	22
Using the Insert Commands.....	22
Using Drag and Drop in Container Fields.....	24
Using Copy and Paste in Container Fields.....	24
Removing Data from Managed Container Fields	24
Compressing Stored Files.....	24
Providing ODBC Compatibility	26
Using Container Functions	26
Base64Decode() and Base64Encode().....	26
GetContainerAttribute()	26
GetHeight() and GetWidth()	27
GetThumbnail()	27
VerifyContainer()	27
Using GetAsText() with Container Fields.....	28
Controlling Container Fields	28
Using the Insert File Script Step.....	28
Using the Insert from URL script step.....	30
Other Script Steps with Enhanced Containers	31
Validating By File Size	31
Installing and Updating Plug-ins with Container Fields	32
FileMaker Go and Enhanced Containers	33
Insert from Device	35
AV Playback Interaction	35
About The Author	36

About this Technical Brief

This technical brief is designed to explain the enhanced container fields feature in the FileMaker product line. In previous FileMaker versions prior to 12, container fields could store files by embedding them or by storing a reference to them. In version 12 and greater container fields can:

- store files externally
- encrypt external files
- create thumbnails
- stream audio and video data
- display interactive content

Prior to the introduction of enhanced container fields, some of these features were available to intermediate and advanced developers through the use of third-party products working in conjunction with native functionality. Beginning developers were often limited to simply embedding files or file references in the container fields.

All developers now will find information of interest to them in this document. This technical brief contains theory and explanations of how files are moved, stored, compressed and delivered back to the user. It also includes step-by-step instructions for setting up databases to use these enhanced container field features.

Understanding Document Management (DM) and Digital Asset Management (DAM)

Long before computers were in wide use, businesses and other organizations had paperwork to manage. Computers were brought into many businesses to convert analog papers to digital. But the promise of the “paperless office” was pushed aside in the wake of many new document types— word processing documents, graphics, video, audio. And so the concept of Document Management (DM) and Digital Asset Management (DAM) arose as developers and users grappled with storing, controlling access to, and sharing digital files. Small or non-profit organizations often manage files without software by creating business rules for naming and manually moving files to a specific location on their server. Larger organizations purchase or create enterprise-level software solutions that control documents throughout their lifecycles from creation to publication to distribution.

Benefits of DM/DAM Systems

Hard cost savings when DM/DAM systems are implemented are dramatic and can be found in reductions for copying, filing and physical storage. The soft returns—employee efficiency and productivity—may be harder to quantify, but are often even more important than hard cost reductions. Some Return On Investment (ROI) calculators for DM/DAM software packages say that up to 30% of employees’ working hours are spent looking for the documents or assets they need. That means that up to 30% of labor costs for many businesses go toward paying people to look for the materials they need before they can perform their actual jobs.

Such large ROIs mean that whatever their overall function, most organizations can benefit from using a DM/DAM system to store digital materials. A court reporting company can reduce or eliminate physical storage of legal transcripts by storing them as ASCII files. Real estate firms can scan or create PDF copies of signed documents to store with their customer data. Companies that deliver printed material by courier or through the mail can create a website that lets their customers log in and retrieve files at will.

Glossary

Container fields can store documents like text, graphic or video files and are the building blocks for any DM/DAM system you create. Though they are created the same way as in earlier versions of FileMaker Pro (select “container” as the field type), options available in FileMaker Pro give container fields more powerful

behaviors and features. Terms for referring to container fields according to their features and a brief description of each follow:

Enhanced Container Field

A container field implemented in FileMaker Pro 12 or greater, which uses interactive, managed, or secure data is an enhanced container field.

Interactive Container Field

Container fields are optimized to display graphics by default. The Inspector lets you optimize a container field for interactive content like audio, video, or PDF. Interactive containers display their content along with an appropriate toolbar for viewing or playback.

Container Field With Managed Storage

You can embed a file in your database or store only a reference to it by selecting the proper option when you insert the file into a container field. But the Storage tab of the **Options for Field** dialog box provides options that let FileMaker Pro manage files by storing them in a location specified in the container field's base directory. This functionality is called *external storage* and files stored in container fields with these options enabled are called *externally stored data*.

Secure Container Field

By default externally stored data is encrypted when it is moved to the base directory. This is called *secure storage*. If you disable secure storage, you are using *open storage*.

Not all organizations will need every feature of a full-blown DM/DAM. But the most common features of DM/DAM systems are listed below:

File check in and check out

Organizations that do collaborative work on files (programming houses or art/design shops, for example) use systems that check files out so they are unavailable to anyone else until they have been checked back in. You can write scripts that prevent container field content from being exported again until the current user has checked the file back into your DM/DAM.

Metadata

Metadata for files includes information like creation date, creator, modification timestamp and modifier. Text files may also have abstracts; image, audio or video files may have keywords for categorization during searches. You can create fields for storing metadata that lets users categorize and find files.

Retrieval

Once a user finds the files they need, the system delivers them to the user. Retrieval systems may include security (access) controls and a checkout system to allow for collaborative work. The Export Field Contents command provides native support for file retrieval and security settings can help determine who has privileges for viewing or exporting files.

Searching

DM/DAM systems give users various methods for finding files. A simple system might provide searches for unique key or file name. More complex retrieval systems let users search by metadata and/or keyword. FileMaker lets you organize managed storage by relationship—a portal could display all files related to a client or project.

Security

Rights management systems control who gets access to stored files and may even gather information about who accesses files, when they are accessed, and if they have been printed or otherwise distributed outside the DM/DAM system. Regulations, like the Health Insurance Portability and Accountability Act (HIPAA), require that documents be protected and access to them controlled. Existing FileMaker Pro security, combined with SSL data transfer, and secure containers' ability to encrypt files stored within the base directory will help you secure digital assets.

Storage

In simple DM/DAMs, files are stored using a predictable system on a centrally located server. In FileMaker, base directories let you use default paths, and Open Storage lets you add a calculation that defines a custom folder hierarchy within the base directory.

Versioning

Systems may keep only the most recent version of a file, or it may include all versions so users can do “rollbacks” or comparisons of different versions. Tables and relationships can help you create a system that ties all the versions of a file to the proper client or project. Fields can store metadata that lets users identify the most recent version of a file. A combination of field options and scripts can enforce integrity of this metadata.

Understanding Container Fields With Managed Storage

The option to “store container data externally” lets FileMaker copy files to a central location. The **Edit > Export Field Contents** command lets users retrieve externally stored data. Database files with managed container fields are smaller than those with embedded files and backups are quicker. Plus, external storage adds the ability to store files securely, in a form that only FileMaker can decrypt.

Note: In FileMaker Go, managed storage is supported only for databases that are hosted by FileMaker Server. See the section, “FileMaker Go and Enhanced Containers,” later in this document for more details on using enhanced container fields in FileMaker Go.

Base Directories on Local Files

When a file is inserted into a managed container field, FileMaker Pro or FileMaker Server copies the file using a *base directory* as part of the storage path. The way in which each application manages its base directories is slightly different, and this section specifically covers base directories when a solution is opened locally in a FileMaker Pro client. Behavior of FileMaker Server base directories is covered in, “Base Directories on Hosted Files,” later in this document.

A database called Design Center, opened in FileMaker Pro, has the default base directory:

```
[database location]/Design Center/
```

The bracketed text is the database’s location on the local computer. When you enable external storage for a container field in the database, FileMaker Pro then creates a directory (folder), which it gives the same name as the database, but without the “.fmp12” file extension.

Note: For databases hosted on FileMaker Server that use external storage, FileMaker Server creates a single base directory, which then holds all of the database-specific subdirectories. Behavior of FileMaker Server base directories is covered in, “Base Directories on Hosted Files,” later in this document.

If you enable Secure storage, then FileMaker Pro creates and manages all additional subdirectories within the base directory. But if you use Open storage, then you have more control over the subdirectories created within the base directory.

By default, when a file is inserted into a container that uses Open storage, the table and managed container names are appended to the base directory to resolve the full path and the file is copied to the new location. Figure 1 shows the full storage path for Design Center.fmp12, which includes a table called “Request” and the managed container field called “FinalProject.”

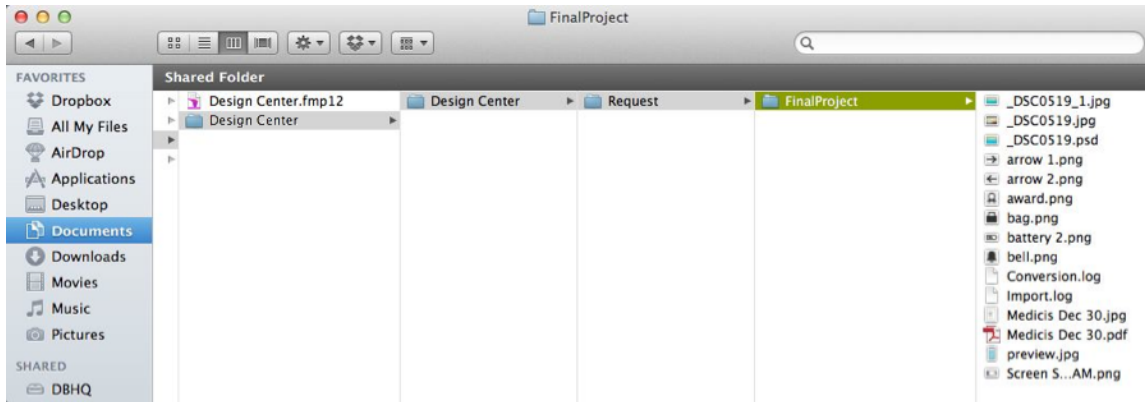


Figure 1. A default base directory for a non-hosted (local) database matches the database file name. By default the full path for an open (non-secure) container field's subdirectories includes the names of the database, table and container field. This path is used to copy files to the new storage location, for display and to remove them when a managed container field is cleared.

If you decide that you need more control over the naming of the table and field subdirectories within the base directory, then you can choose to replace the default Open storage path in **Field Options > Storage** tab with a calculated value (Figure 4).

Managing Base Directories for Local Files

The default base directory can be customized and new ones added if you want to use another location on a local computer or distribute external storage on remote volumes. You must have **Full Access** privileges to manage base directories. Note also that the ability to define base directories on remote volumes is supported only by FileMaker Pro. You will not be able to use remote volumes for base directories in FileMaker Server, which will be discussed later in this document.

To add, edit or delete a file's base directory, choose **File > Manage > Containers** (Figure 2).

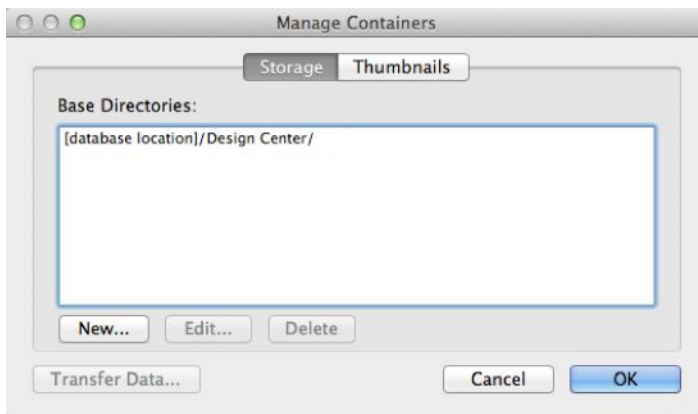


Figure 2. The **Manage Containers** dialog box, as it appears on a local database. The **New**, **Edit** and **Delete** buttons are unavailable when a file is hosted.

On locally stored databases, base directories can be relative, full or network paths. You can customize them even after files have been inserted into the database. But there are some rules to keep in mind:

-
- Base directories cannot refer to a parent directory ("../").

- FileMaker Pro can transfer externally stored data if you edit a base directory. But transferring externally stored data can be time-consuming if your database contains many files. So it is more efficient to set up custom directories when first defining your managed storage strategy rather than letting a potentially lengthy transfer take place after editing a base directory in an already populated database.
- You cannot delete a base directory using Manage Containers if the base directory contains data, if a container field refers to it, or if data transfers are pending. (See the section on Transferring Data Between Storage Types for more information.)
- When a base directory refers to a remote volume, in order to view and insert data in the target container field, users must have the remote volume mounted and appropriate permissions for the remote base directories.
- Base directories cannot be managed when the database is hosted—either from FileMaker Server or peer-to-peer networking.

Adding and Editing Base Directories

The Edit Base Directory and New Base Directory dialog boxes show examples for each directory path format type (Figure 3).

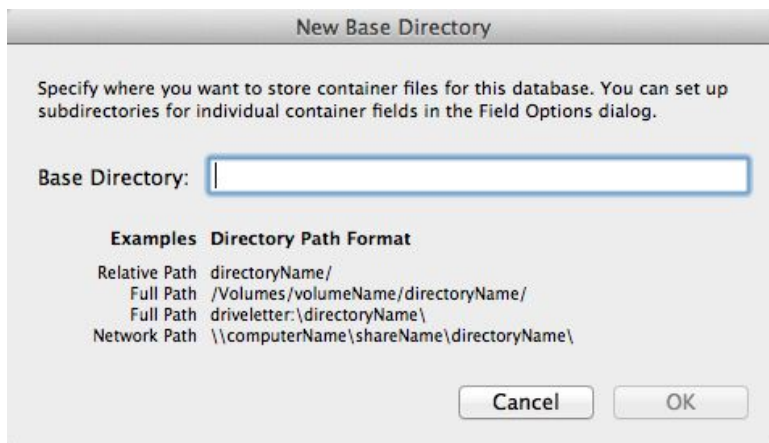


Figure 3. Base directories use existing supported path formats and can be full, relative or network paths. If the base directory path you entered is not valid, a warning appears when you click OK.

If reserved characters or names are used in a base directory, an error message will appear when you click OK to confirm the changes. Reserved characters are:

- < (less than)
- > (greater than)
- \ (backslash – as part of a directory name)
- / (forward slash – as part of a directory name)
- | (vertical bar or pipe)
- * (asterisk)
- ? (question mark)
- “ (double quotation)

Normally a colon should not be used as a part of a subdirectory name. But if you are using open storage with calculated relative paths, you can rename files as they are copied by ending the base directory with a

colon (:). A relative directory “/Design Center/req_.” will store files in the Design Center directory and rename them, adding the prefix “req_”.

These characters can cause ambiguity in path names and are best avoided:

- [and] (left and right brackets)
- = (equals sign)
- + (plus sign)
- % (percent sign)
- \$ (dollar sign)
- , (comma)

Note: Take care when creating custom base directories. While you could create directories in multiple locations on a local computer or network, that practice leads to unnecessary complexity. If the automated methods for moving databases and their external data cannot be used, it can be tedious to move files manually and the process is prone to error. Creating custom base directories is best used for shortening or making schema names conform to OS limitations.

Specifying Base Directories for Managed Container Fields

Managed container data storage is centralized by default, but you can set each managed container field to use a different base directory. To manage volume space or to control direct access to the external files, you can create multiple base directories and then set individual container fields to work with an appropriate directory (Figure 4).

The screenshot shows the 'Options for Field "Photographs"' dialog box with the 'Storage' tab selected. The 'Global Storage' section has a checkbox 'Use global storage (one value for all records)' which is unchecked. The 'Repeating' section has a text box 'Maximum number of repetitions:' with the value '1'. The 'Container' section has a checked checkbox 'Store container data externally'. Below it, the 'relative to:' dropdown menu is open, showing a list of base directories with '[database location]/DesignCenter/' selected. There are also radio buttons for 'Secure storage' (unchecked) and 'Open storage:' (checked), with the 'Open storage:' text box containing 'Request/FinalProject/'. A 'Specify...' button is next to the 'Open storage:' text box. At the bottom are 'Cancel' and 'OK' buttons.

Figure 4. The Storage panel of the **Options for Field** dialog box has a section for managing external storage. The “relative to” pop-up menu shows a list of all base directories for a database.

Base Directories on Hosted Files

When you move a database that uses managed containers to FileMaker Server, the default base directory shown in the Manage Containers and **Options for Field** dialog boxes is:

`[hosted location]/Files/databaseName/`

When FileMaker Server is installed, a directory named “RC_Data_FMS” is created in the directory where the databases are stored. As databases that use managed containers are uploaded to the server, their base directories are converted relative to RC_Data_FMS (Figure 5).

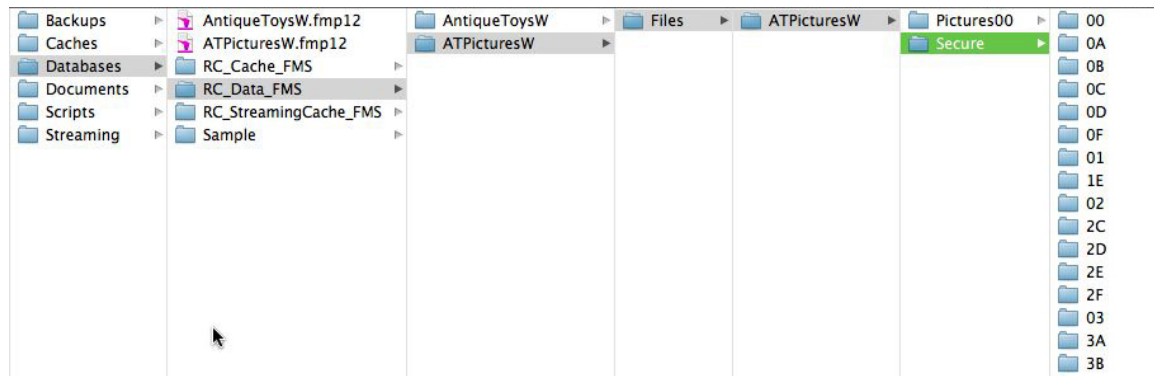


Figure 5. The base directories for databases hosted on FileMaker Server are converted relative to /FileMaker Server/Data/Databases/RC_Data_FMS/. This figure shows a database (“ATPicturesW”) with an open storage container field named, “Pictures00” and a secure storage container field denoted by the placeholder, “Secure”.

Each database will have a full directory inside RC_Data_FMS/. But for more separation, you can create subdirectories in /FileMaker Server/Data/Databases/ as shown with the installation of FileMaker Server sample database (Figure 6).

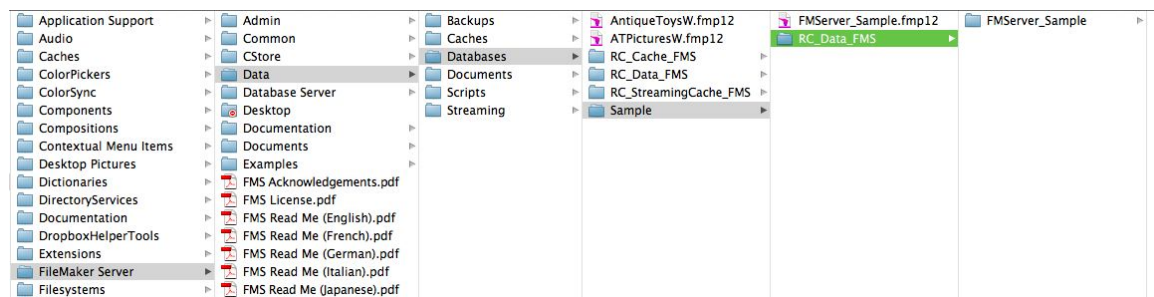


Figure 6. If you create subdirectories in /FileMaker Server/Data/Databases/, each one will have its own RC_Data_FMS/ for externally stored data.

Note: If you have configured FileMaker Server to use additional database folders, then you can also specify a different folder in which to externally store container field data. See the section, *Configuring Database Server settings > Database and backup folder settings > Setting up the container data folders* in [FileMaker Server Help](#) for more information.

Understanding Managed Container Storage Types

Managed container fields use secure storage by default, but open storage can be chosen instead. When a mix of secure and open storage is combined with custom base directories, developers will be able to create DM/DAM solutions that match a variety of business rule and IT needs.

Secure Storage

Secure storage containers encrypt data using Advanced Encryption Standard (AES-128), the standard used by the US Government National Security Agency (NSA) to protect classified information, up to SECRET level.

Note: If you have used database encryption in FileMaker Pro Advanced to encrypt your database file, then any externally stored container data will also be encrypted using AES-256 instead of AES-128. You can find out more about encrypting databases in [FileMaker Pro Advanced Help](#).

A randomly generated encryption key is assigned to the container for each record. The key is used to encrypt a file inserted into the container field. After encryption, secure container files cannot be opened directly by the application that created them. To edit them, encrypted files must be decrypted by FileMaker Pro using **Edit > Export Field Contents**. Use the original application to make changes to the file and then reinsert the edited file so FileMaker can re-encrypt it.

A 32-character MD5 (message-digest algorithm used to verify data integrity), of the container content forms the directory structure for storing secure container data (Figure 7).

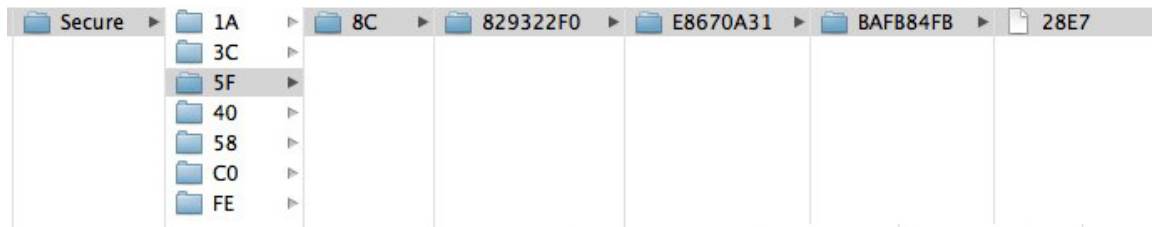


Figure 7. Secure container data is stored using a 32-character MD5 of the container content that is broken up to form the folder structure. This folder structure persists if externally stored data is cleared from the container field. New data placed in the field uses the same folder structure as the original file.

A cryptographic MD5 hash of each externally stored file is tracked in the database. When secure container data is requested, the hash is used to check the file for consistency before it is displayed. If the hash does not match, the file is considered “tampered with,” and FileMaker Pro and Go will not display it (Figure 8). If access is allowed to them, secure storage directories or files can be renamed or deleted, but containers will not display data that has been changed after being inserted into a managed container field.

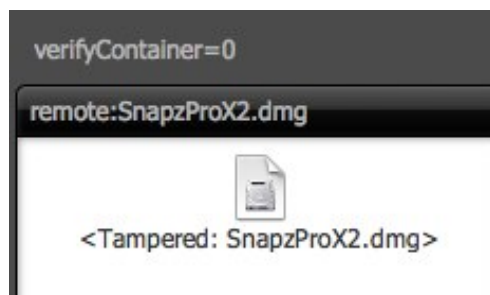


Figure 8. FileMaker Pro does not display or export tampered files. The file label of this secure container displays the text “<Tampered:” followed by the file name. A merge variable has been placed on the layout above the container field to display the result of a VerifyContainer() function (verifyContainer=0).

Cached container data is not checked for consistency, but you can use the `VerifyContainer()` function to enforce a check, whether the data is in cache or not. `VerifyContainer()` returns a value of 0 (Boolean false) when external container data has been changed or deleted. Unchanged data returns a value of 1 (Boolean true).

Note: Available RAM on the client computer limits the size of encrypted files. Control the size of inserted files by setting a maximum file size on the Validation panel of a container field's options.

Using Open Storage

If security is not an issue, you can change the container field default option of Secure storage to Open storage. Like secure storage, open container data is copied automatically using the field's base directory, but is not encrypted. Files retain their original formats and can be opened and edited without being exported from FileMaker Pro first (Figure 9). However, if files are edited from their external storage directory, they will be considered tampered with and will not display.

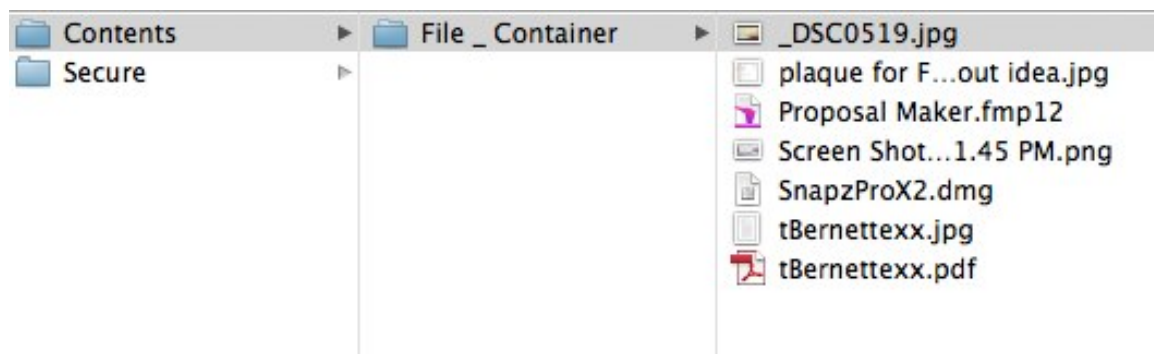


Figure 9. The default folder structure for open containers is easily legible, and can be customized using the calculation engine. Open container files are stored in their native format and could be edited without being exported from FileMaker Pro first, so access to these directories should be restricted.

Best practice for databases hosted on FileMaker Server has long been to avoid file sharing for files in the `/FileMaker Server/Data/Databases/` directory. This recommendation is equally important for externally stored files and is now extended to externally stored files on peer-to-peer hosted and standalone databases. Files inserted into an externally stored container field be managed by FileMaker Pro; they should be exported using the **Export Field Contents** command before they are edited by their native applications.

Note: Editing files within the base directory can also cause problems with progressive backups in FileMaker Server 13 and above (see the section on Progressive Backups and Container Fields)

Managing Open Storage Paths

The Storage panel of the field options dialog box (refer back to Figure 4) also shows the default table and field name subdirectories that are added to the base directory to complete the container's storage path. You can customize this part of the path using static data, but like customizing the base directory, this practice is best left for simplifying or shortening schema names. Click the **Specify** button for access to the calculation engine. Human-readable data can be added into the folder structure by using the calculation engine to create directory names, with FileMaker Pro primary key values, for example. Or customer or project names can be included if key values have no business meaning or would compromise security because they are visible in the directory.

Open storage calculations are resolved when the container data is inserted and the result is stored in the container. You can see calculation results, along with other metadata, with a `GetAsText ()` function that uses the container field as its parameter.

Sample results of the `GetAsText` function are:

```
remote:puppet.jpg
size:761,708
DPI:300
JPEG:Pictures00/toyContainer/toyID_65734/puppet.jpg
size:1920,1200
image:../ Figure 5.png
imagemac:/MacIntosh HD/Writing/TechBrief/Figure 5.png

remote:Clip 1-9-13 at 6:46 PM.mov
MOV :Videos12/January/Clip 1-9-13 at 6:46 PM.mov
remote:Conversion.log
FILE:Secure/CC/CC/73669612/DA997CB0/91CADB13/532A
```

Subdirectory names in open storage paths are not updated if the database or schema is renamed. But since the original path is stored in the container field the calculation resolves correctly and container data stays linked. However this does not work in reverse. If subdirectories are renamed or moved, external containers will display <Missing:theFileName> errors. Restore normal display of external container data by:

- replacing directories if they have been moved, or;
- restoring original directory names.

View the original directory names using a `GetAsText ()` function, passing the external container as the parameter.

Transferring Externally Stored Files

When pre-12 databases are converted, container fields are not automatically changed to managed storage because container fields in FileMaker Pro versions greater than 12 can still store files as embedded or by reference only. Storage options can be changed after data has been inserted into containers with no risk of loss of data. This means that embedded files can be converted to external storage and open containers can be changed to secure storage (and vice versa).

Choose a storage type for a container field with the **Options for Field** dialog box. The options you need are in the Storage panel (Figure 4). When a container's storage type, base directory or open storage path is changed, the **Container Data Transfer** dialog box (Figure 10) appears. Because transfers can take time for files with many large externally stored files, the dialog box gives you the choice to close it without transferring the data immediately.

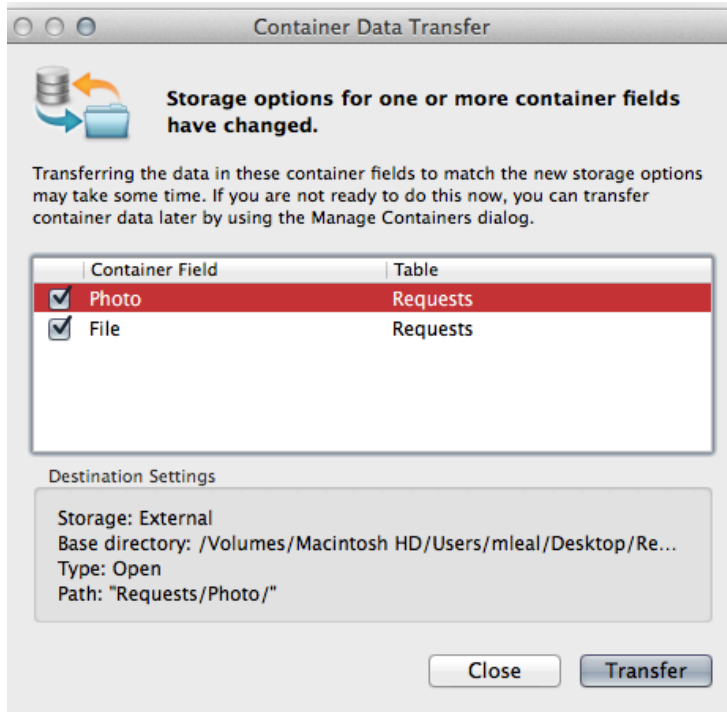


Figure 10. The **Container Data Transfer** dialog box lists the managed container fields that need their data transferred. To break up the transfer of data into multiple sessions, you can deselect one or more fields, click the **Transfer** button and then return to this dialog box later to transfer other field data.

If you close the **Container Data Transfer** dialog box without transferring data, you can view it again. Choose **File > Manage > Containers** (Figure 2). If any managed container fields have not had data transferred, the **Transfer Data** button is enabled. To transfer the external data:

1. Click the **Transfer Data** button. The **Container Data Transfer** dialog box appears, showing a list of container fields that have not been transferred. All fields are selected for transfer by default.
2. Click the checkbox to the left of any field to deselect it. Only checked fields will be transferred.
3. Click the **Transfer** button. FileMaker changes the storage type and moves the files to the new storage location.

Files are protected while being transferred in two ways. First, the conversion process saves each container with its new options individually. Conversion is transactional, so if any file cannot be converted, the transaction is not committed and the old data is retained. Second, container fields hold multiple versions of a file while they are being edited so they can support the **Undo** and **Revert Record** commands. That is, when a new file is inserted into a container field, the old data is not deleted until the record is either committed or reverted. This “stacking” of files within a container also serves as protection for the files during conversion since the original file is not deleted from each container until the new data has been written to disk. So even if disk space runs out during a conversion, container data is not at risk.

When externally stored data is moved, encrypted or decrypted, a **Transfer Summary** dialog box appears after each transfer operation (Figure 11).

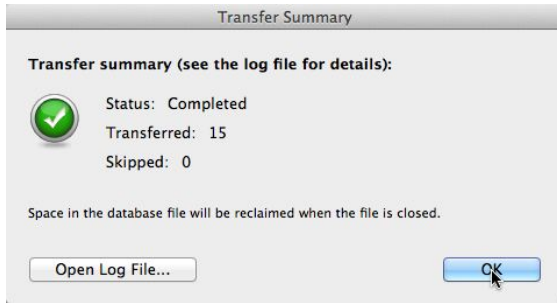


Figure 11. This **Transfer Summary** dialog box appeared after an open storage field was converted to secure storage. In this transfer, all items were converted successfully. For more detail, as when items are skipped or to check the time taken to transfer or convert the data, click the **Open Log File** button.

A Transfer.log file is created after the transfer (Figure 12). If container data cannot be transferred or converted, the log gives details about which records were skipped.

Note: If a record is locked because a user is editing it when you transfer data, its external data cannot be transferred. However FileMaker Pro tracks the data that was not transferred, so you can transfer data in two batches. First transfer the data to process all the unlocked records. Then you can disconnect clients briefly to transfer the records that were locked during the initial transfer.

DATE	FILE	RESULTS
2012-03-19 15:57:48.229 -0700	ATPictures.fmp12	Container Data Transfer started.
2012-03-19 16:11:02.155 -0700	Warning	[Record ID 45767, field Pictures00::Thumb2] Skipped non-transferrable data: File reference or text
2012-03-19 16:11:06.017 -0700	ATPictures.fmp12	Transfer Summary: status: Completed; errors: 0; items transferred: 2585; items skipped: 1.
2012-03-19 16:11:06.018 -0700	ATPictures.fmp12	Container Data Transfer ended.

Figure 12. This sample Transfer log has been formatted as a table for legibility. In this log, a pre-12 database was converted and then a container field with embedded data was changed to managed storage. The second line shows a warning, where one item, a file inserted with a reference only, was not transferred. The Record ID is provided so you can find and process the record manually. First create a calculation field with the formula **Get(RecordID)**. You can search the calculation field for the ID you need, export the referenced file and then re-insert it as external data.

Note: Since converting embedded container fields to external storage removes large files from a database, this type of transfer can have a dramatic impact on file size. On a local file, unused space in the file will be reclaimed when the database is closed.

Because of risk of data loss, some types of data stored in container fields are not transferred when storage types are changed. They are:

- text pasted into container fields
- layout objects pasted into container fields
- sound files recorded natively in FileMaker
- Microsoft OLE objects
- files inserted by reference

If you want to convert text objects to container managed storage data, choose **Edit > Export Field Contents** to export the pasted text, which converts it to a file that can be reinserted as external data.

Because Microsoft OLE is deprecated (see Section 11.2 of the FileMaker 11 ReadMe file, http://help.filemaker.com/app/answers/detail/a_id/10007), you should launch embedded OLE objects into the parent application and save the document in its native format. Once the document is saved in its native format, you will be able to re-insert it into container fields again.

Files inserted by reference can be reinserted as external data using the following 2-step FileMaker script:

```
Set Variable [ $path; GetAsText (referencedContainer) ]
Insert File [ Insert; referencedContainerField ; $path ]
```

Handling Large Files

Many DM systems store text or spreadsheet documents, which are typically small enough to upload and download very quickly. But photographs, even when saved in formats like jpeg, can be very large and may take some time to move from host to server. Other digital assets, like the audio from a music studio, or the video files created by a film production company, where even a three-minute HD video clip can be 100 megabytes or more, can dramatically increase the time needed for moving large files quickly across a network. FileMaker Server uses several strategies and technologies for making even large files display quickly in container fields.

Thumbnails

Instead of transferring a large image over the network, where it might be reduced to fit into a very small container field, FileMaker Pro creates thumbnails and displays them instead. By default thumbnail generation is turned on and thumbnails are stored in cache, which is emptied when the database is closed. Thumbnails are created on the server and then transferred over the network to the client, but are cached on both the server and the client for better performance.

To speed up display, thumbnail storage can be made permanent on the host. In the **Manage Containers** dialog box, click the **Thumbnail** tab and select **Permanent storage** (Figure 13). In this case, thumbnails are generated and then written to disk. These permanent thumbnails survive server shutdown or crash, and recently used thumbnails are kept in cache for quick access. Permanent thumbnail cache is on-disk and in-memory. Disk cache is saved when the database is closed.

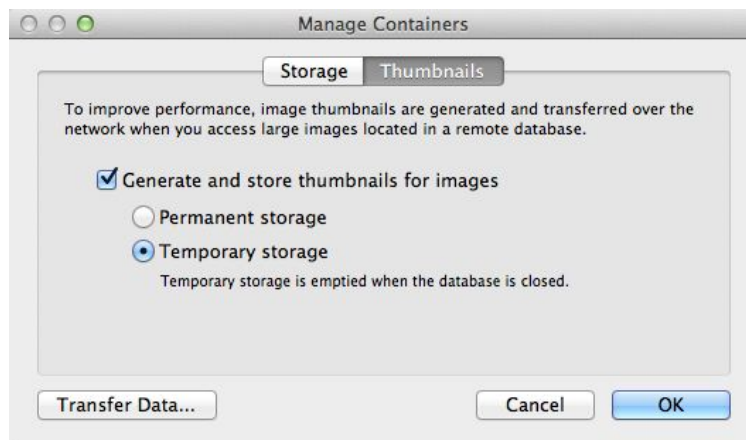


Figure 13. The Thumbnails panel of the **Manage Containers** dialog box lets you turn off the default option to generate thumbnails or make thumbnail storage permanent.

Note: Thumbnails for secure container images are encrypted.

Thumbnails, whether they are permanent or temporary, are created by reducing the original image by a factor of two until the smallest rectangle that is larger than the container is reached (Figure 14).

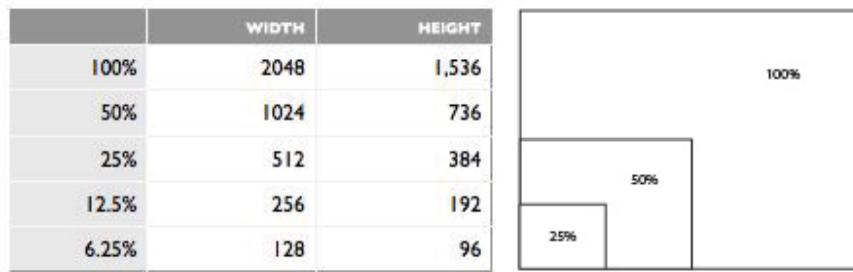


Figure 14. Thumbnails are reduced from the original image's size but are always the same proportion as the original image.

Note: The `GetThumbnail()` function generates a thumbnail of a specific size. See the section on Using Container Functions for more information.

Streaming Data

Audio, video and PDF files can be streamed from FileMaker Server to interactive container fields using Apache Tomcat—an open source Java Servlet engine that is used as a standalone web server or along with another server package, like Apache or IIS. Tomcat and Java are automatically installed with FileMaker Server.

Note: Tomcat streams data via port 80 and 443, so they must be opened in systems where firewalls sit between the server and clients.

Streaming data files are buffered to the container and can automatically start playing when the buffer is filled instead of waiting until the entire content has been downloaded. Streaming also allows users to jump forward or backward to any point in the playback without waiting for the buffer to fill up again. Since it is a separate process from FileMaker Server, Tomcat streaming also frees up resources for record data.

Data is streamed when these conditions are met:

- Data format stored is either PDF, audio or video files.
- The container field layout object must be optimized for “Interactive content” in the Inspector’s Data tab.

Optimizing Container Fields

In pre-12 FileMaker versions, interactivity with a file was controlled by the command used to insert it into a container field. For example, you could choose **Insert > Picture** to place a JPEG in a field called “Requests” and an image was displayed. **Insert > QuickTime**, used with a PDF, displayed the file and a controller for scrolling through the document’s pages (Mac only).

In FileMaker Pro 12 and greater, you control content interactivity by selecting the appropriate optimization for the field object. Because interactivity is provided through web browser technologies, it is supported cross-platform. When you convert a database that stores images and PDFs, image files are displayed as thumbnails, but PDFs are displayed as file icons and are not interactive. But a second layout, using a field object that is optimized for interactive content solves the problem (Figure 15).

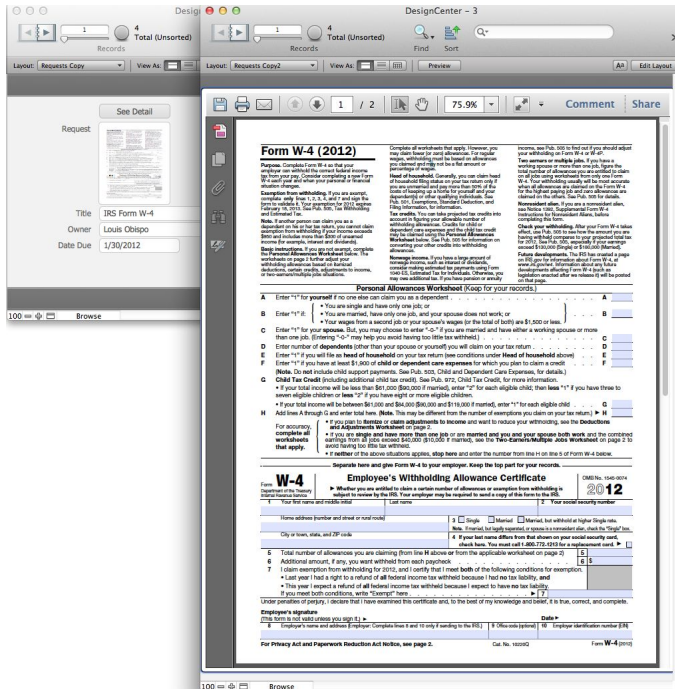


Figure 15. To provide interactivity in converted databases, create a button that runs a script to detect the file type—`GetAsText (Design Center::Requests)` is the starting point for parsing the file extension—and then navigate to a layout that has a field object that is optimized appropriately for the file. See the section on *Writing Scripts for Enhanced Containers* for more information on controlling how users insert files in FileMaker Pro.

Optimize for Images

The default optimization option for container fields (“images (JPEG, PNG, BMP)”), gives them the same behavior as pre-12 containers—they are optimized for displaying image files. When graphic files are inserted with the **Insert File** command, a standard file icon displays with the file’s name. When they are inserted with the **Insert Picture** command, a thumbnail of the graphic is generated for display instead.

Optimize for Interactive Content

To format a container field to display streaming data, choose the **Interactive content** option on the Inspector’s Data tab (Figure 16).

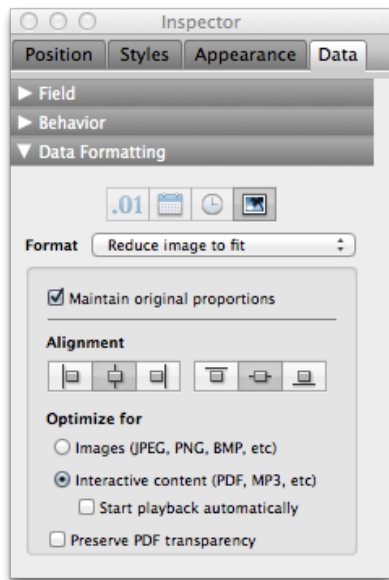
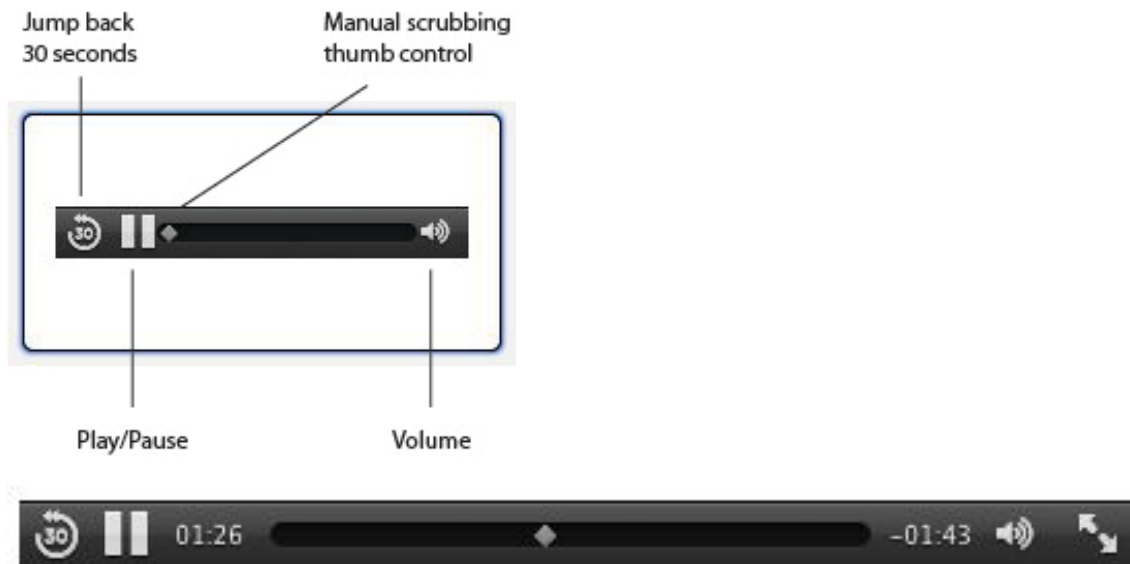


Figure 16. The Data Formatting section of the Data tab of the Inspector provides options to optimize container fields for streaming data and PDF controls.

When you have selected the Optimize for “Interactive content (PDF, MP3, etc)” option, the “Start playback automatically option” is also available. When that is selected, users do not have to click the play button for audio and video files. The data is requested when a record is loaded and the file plays automatically when enough data is buffered to the client.

Appropriate player controls are displayed when audio and video files are inserted into an interactive container field (Figure 17).



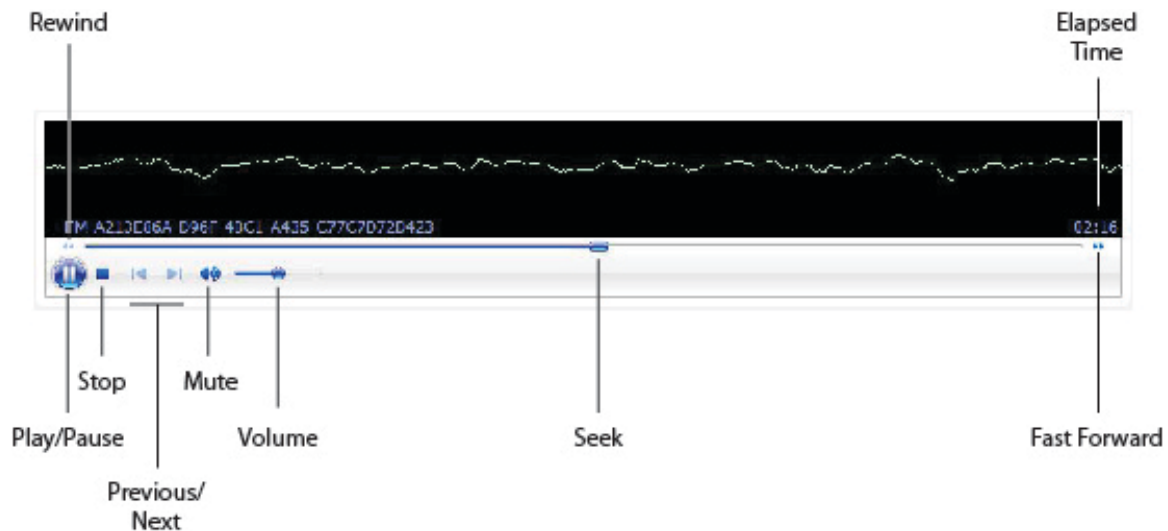


Figure 17. Top. The Mac audio controller for an interactive container field (shown with active field highlight) lets users start or pause playback, scrub through the file, jump back 30 seconds and control volume.

Middle. The Mac video controller has all the controls of the audio controller, plus it shows elapsed and remaining times on its progress bar. The diagonal arrows zoom the video for full screen playback.

Bottom. The Windows audio controller has tools to rewind, fast forward, play/pause, stop, mute, volume, seek and a display of elapsed time. A waveform for the audio file is displayed above the controller. The video controller has the same tools, but shows the video instead of the waveform.

Native Format PDF

PDF documents are downloaded to the local hard drive where they are placed into web container cache and displayed. The full suite of web container PDF tools is available. Because some tools are accessible only in a slide-out panel, interactive field objects that support PDFs may need to be larger than a full size page if you want to display them at actual size. With interactive PDFs, you can:

- scroll or click through pages
- zoom
- select and copy blocks of text or the entire PDF document

Other features are available on Mac OSX, when Adobe PDF Reader is installed:

- print
- attach the file to an email or use Adobe SendNow Online

Note: To insert and view PDFs as streaming content on Windows, a PDF-capable web browser plug-in (like Adobe Reader) must be installed on the client machine. If a container field is set to be optimized for interactive content, but the **Insert > PDF** command is not available, the plug-in may be missing.

Distributing Server Requests

FileMaker Server is multi-threaded. That means it can handle several threads, or tasks, at one time by passing new tasks to other processors. When all the server's processors are busy, new tasks are sent to a job queue, where they wait for the first free processor. So if you will be storing lots of data that requires streaming, or even if the server will be handling lots of large graphic files, the more processors your server has, the better performance will be.

Understanding Data Cache

Since FileMaker Server can host up to 125 databases, separate memory cannot be allocated to each database. A cache dispenser controls shared memory and then distributes it to individual caches as data is requested. When memory is full, the cache dispenser clears the oldest cached item to make room for incoming requests.

Streaming data has its own cache directory at

/FileMaker Server/Data/Databases/RC_StreamingCache_FMS. Audio, video and PDF data is cached as it is inserted or requested by an interactive container field. A hard link to the file is created in the streaming cache folder when a file is inserted or requested. Embedded files are exported to this folder before a hard link can be created for them so they can be cached for streaming. And secure containers are decrypted into this container when they are requested. Streaming cache is cleaned up every two hours.

Many resources (multiple threads, multiple caches, multiple databases) compete for server memory, so cached items are locked while they are in use so they are not removed from cache. If an item that has been locked in cache is deleted, it is not cleared from external storage until the last lock is released.

Saving Volume Space

When the same file is inserted into container fields in multiple records, it will be reused where possible. The file itself must be an exact match—the base directory must match and the storage types must agree. For example, if a file is inserted in an open container and in a secure container, the two files do not match and so two copies of the file, one secure and one open, will be stored. Or if you insert the same file in two different container fields that use open storage in the same record, the default open storage paths are different, and so two copies of the file are stored.

Moving Databases with Managed Storage

A database with externally stored data can be moved between volumes with its links and data intact. The move can be done manually, but FileMaker Pro and FileMaker Server each provide a method that is easier and more reliable. The menu, **File > Sharing > Upload to FileMaker Server** is the preferred method for moving a database to FileMaker Server. If the destination is a standalone computer or an iOS device, FileMaker Pro can save a copy of the database and its externally stored data in self-contained file that can be safely moved from volume to volume.

Uploading a File to FileMaker Server

You can upload a database file to FileMaker Server directly from FileMaker Pro by choosing the menu, **File > Sharing > Upload to FileMaker Server**. This command first prompts you to authenticate with FileMaker Server and then copies the database and its external data to the proper directories and sets appropriate file properties (permissions). For large databases with many external files, this process may take several minutes. A progress bar and text feedback informs you about file preparation and upload.

Note: Externally stored data is not culled during upload, so if there are stray files in the base directories, they will be uploaded to the server.

To learn more about uploading files to FileMaker Server see [FileMaker Pro Help](#).

Saving a Self-contained Copy of a Database

If a database that uses containers with managed storage must be moved from one “standalone” installation to another, you can save the database as a self-contained copy. This process embeds externally stored container field data into the database, which can then be moved as a single file. A standalone database may be used:

- on an iOS device with FileMaker Go
- on a single computer
- hosted by FileMaker Pro on a peer-to-peer network

The iPhone and iPad cannot store container data externally, so the database will be used with its embedded data on those devices (See the section on FileMaker Go and Enhanced Containers for more information). But a file that you intend to use on a single computer or on a peer-to-peer network can be converted back to external container storage on the target computer.

To save a database, its externally stored data, and data stored as a file reference, as a standalone file:

1. If the database is hosted, disconnect all clients and then open it with FileMaker Pro or FileMaker Pro Advanced.

As with other options in the **Save As** dialog box, hosted files cannot be saved as self-contained copies. In a multi-file solution where not all files have managed containers, only the files with managed containers need to be saved as self-contained copies. Saving files without managed containers as self-contained has no effect on them except to make a copy.

2. Choose **File > Save a Copy As** and then select the “self-contained copy (single file)” option. Click the **Save** button.

External data is embedded into the database’s container fields for safe transfer.


The option to **Store container data externally** is turned off in the copy but any custom settings you made in each field’s relative directory are kept. When the file has been copied to its target location, use the **Options for Field** dialog box (Storage tab) to select **Store container data externally** for each field. When you close the **Manage Database** dialog box, the **Container Data Transfer** dialog box appears. When you click **Transfer**, the embedded files are moved to base directories on the new volume.

Learn more about saving and copying database files in [FileMaker Pro Help](#).

Downloading Databases with Managed Containers

When you need a local copy of a hosted database from FileMaker Server, use the **Download Database** command in the Server Admin Console, which moves databases and the associated base directories to a location you specify. You can download files to any network drive—just mount it on the machine on which you’re using the FileMaker Server Admin Console before you start the download. Databases and the base directories containing the external container files are compressed into a single .zip file for easy download. Unzipping the file gives you the downloaded database(s) and base directory(s) for the managed containers.

To download a database and its externally stored data:

1. In FileMaker Server Admin Console, choose the Activity > Databases panel and then close the databases you want to download. Open databases cannot be downloaded.
2. In the  pop-up menu, choose **Download Database**.
3. For each database you choose to download, FileMaker Server 14 creates a .zip file containing the database and its external storage base directory. Each .zip file is named after the database it contains and is saved to the download folder selected in your web browser’s preferences.

Backups and Managed Containers

FileMaker Server scheduled backups use a storage strategy called “hard links,” that are faster and can save hard drive space when databases or their externally stored files don’t change between backups. This has dramatic implications, especially in large DM/DAM systems where thousands of external data files are stored. Hourly backups of externally stored data could quickly fill even a large hard drive without hard link technology.

Understanding Hard Links

Hard links let multiple directory entries refer to a single file. The concept is similar to file aliases, which are pointers to another file that let you access the file without knowing its exact location. But unlike aliases, hard links are actually the same instance of the file they are linked to. Any change to one hard link entry affects “its” file and all other hard links to the file.

During scheduled backups, the first backup for a schedule is a complete copy of data (actual files). Each subsequent backup for that schedule copies only the files that have been edited, added or deleted, and creates hard links to the unchanged files. This means subsequent backups use only a fraction of the space needed by the original and complete faster (Figure 19).

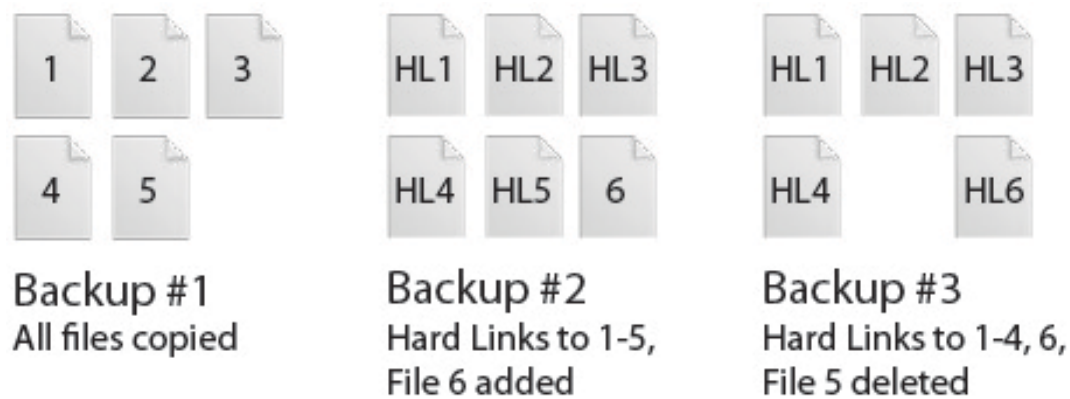


Figure 18. This diagram shows how hard links are used to create FileMaker Server scheduled backups. The first backup is a complete copy of all files (both the databases and their externally stored data). Subsequent backups copy only the files that have changed. Since all hard links to a file are really the same file, editing files in a backup folder will change all the backups for a schedule.

Because each hard link is the same instance of a file, deleting a hard link from a backup does not delete the file itself because previous backups will have a hard link for the file. This makes files extremely durable, in the sense that deleted files are easily recovered by restoring a previous backup.

However this durability of files also means that changing a managed container file in any backup folder changes the file in *all* the backups. This is another reason, as previously discussed in the section on open storage, why you should implement a policy requiring files to be exported from their container fields before they are edited. Letting FileMaker Pro manage externally stored data will let you “roll back” to a previous version of a database, and its externally stored data, by restoring the appropriate backup.

Working With Enhanced Container Fields

In FileMaker Pro, container fields support embedded, referenced and external data storage for files with different formats and display/playback needs. As in previous versions, the **Insert** command dialog boxes let you choose specific options, like storing a reference to the file or compressing it before it is embedded. Enhanced container fields also support drag and drop and copy-paste (see the Using Drag and Drop in Container Fields, and Using Copy and Paste in Container Fields sections below for more details).

Using the Insert Commands

To support multiple file formats and to provide backward compatibility, FileMaker Pro has six menu commands for inserting files into container fields:

- Picture
- QuickTime

- Audio/Video
- PDF
- File

The Picture and File commands are available for all containers, but the Optimize setting on a container field's layout object determines which of the other commands are available. See Table 1.

Container Field Optimization	Available Commands
Images	Picture, QuickTime, File
Interactive Content	Picture, Audio/Video, PDF, File

Table 1. If the command you want is not available, check the container field's optimization setting in the Inspector's Data tab. The field's storage type has no bearing on the available Insert commands.

Choose an **Insert** command based on the format of the file you are inserting and the options you need. For example, even if a container field is defined to store container data externally, if you need to store only a reference to a file you can do so when you insert it. But not all Insert commands give you the option to store only a reference to a file; only the Picture, Audio/Video, PDF and File commands have this option. When a file reference is inserted, the file itself is not moved to a base directory or embedded into the container. As in previous versions of FileMaker Pro, files stored as references are not available to other users of shared databases unless the file is on a shared volume that is mounted on the local system. Referenced files cannot be displayed if they are moved, renamed or deleted. See Table 2.

Command	Formats	Options	Results
Picture	EPS, GIF, JPEG, PICT, BMP, FlashPix Image, JFIF, JPEG 2000, MacPaint, Photoshop, Picture, PNG, PDF, QuickTime Image, SGI, Targa, TIFF, WMF*, EMF	Store only a reference to the file	The file's content is displayed.
QuickTime	QuickTime **, AVI, Cubic VR, DV, FLC, Karaoke, Flash 5, MPEG, PDF, QTM, VR	N/A—this command always stores only a reference to the file.	QuickTime controls allow the movie to be played.
Audio/Video	AIFF, AVI, MP3, MPEG-4 (audio and video), MPEG, MOV, QT, Sun Audio, Wave Audio, WMA, WMV	Video, Audio Store only a reference to the file	The file can be streamed, with or without the option for autoplay.
Sound	AIFF, AU, Audio CD Data, MP3, SoundFont2, SND, WAV	N/A	A sound is recorded by FileMaker and/or can be played using QuickTime controls.
PDF	PDF	Store only a reference to the file	PDF controls allow scrolling through the document, zooming and copying of text.
File	All file formats	Store only a reference to the file Compress	A file icon and the file name are displayed.

*Table 2. If the file format or options you want are not available, you may need to choose a different **Insert** command.*

* On the Mac, with the exception of WMF format files, Flip4Mac must be installed.

** Some formats, like QTVR, are not supported.

Note: On Windows PDF and QuickTime files require supporting browser plug-ins. If either command is not available, or if inserted content is not displaying properly, install the appropriate plug-in and then restart FileMaker Pro.

Using Drag and Drop in Container Fields

FileMaker Pro supports drag and drop insertion of files on Windows and Mac. Files can be dragged from the desktop or from one container field to another. Note that drag and drop between container fields is supported only when the target is not set to be interactive. Text can be dragged from other applications. If you drag and drop into a container that already has a file in it, the dragged content replaces the container's contents.

Likewise, drag and drop inserts data without letting you choose options. An image file is inserted as a picture and a word processing document is inserted as a file. Audio, video and PDF files can be dragged to a field optimized for images, but only the file name and a file icon or thumbnail display in the container. These files can be played in another instance of the field that is optimized for interactive content.

The container field's storage type is the main controller of how dragged data is stored, but the same rules as for transferring data apply to such data. That is, if you drag a file into a managed container field, the file is moved into the base directory. But if you drag text, whether from another FileMaker field or from another application, the data is embedded.

Using Copy and Paste in Container Fields

Copy and paste into a container field is supported and follows the same rules as dragged data above. You can also copy-paste graphic objects from some applications into a container field.

Note: FileMaker native layout objects that are pasted into interactive containers do not display. But a version of the field that's optimized for images will display the data.

Removing Data from Managed Container Fields

When a file is removed from a managed container field, the external file is removed from the base directory when the record is committed. Remove externally stored data from a managed container by highlighting the field and then choosing the menu, **Edit > Clear**. **Edit > Cut** can also be used, but since it puts the data on the clipboard, this command is best used if you want to paste the data after removing it from the container field. When the field is optimized for images you can also hit the forward or backward delete key. Only forward deletion is supported for interactive fields.

Compressing Stored Files

Files can be compressed as they are inserted into a container field. Compression saves space in base directories and can be especially useful for managing file size when the data is embedded in a database. But compressed files cannot be streamed or displayed as pictures; the option to compress a file is available only through the menu, **Insert > File**.

Files are compressed by FileMaker Pro and then moved to the base directory, even when the database is hosted on FileMaker Server. Compressed files are stored in .gz format on both Windows and Mac. This format uses the same algorithm as ZIP compression, but yields higher compression rates.

To compress a file as it is inserted into a container field:

1. Select **Insert > File** and then choose the file you want to insert into the container.
The compress option is only available when data is inserted with the **Insert > File** command.
2. Choose the **Compress** option (Figure 19).
3. The file is compressed as it is inserted into the container. A progress dialog box may appear while the file is being compressed (Figure 20).

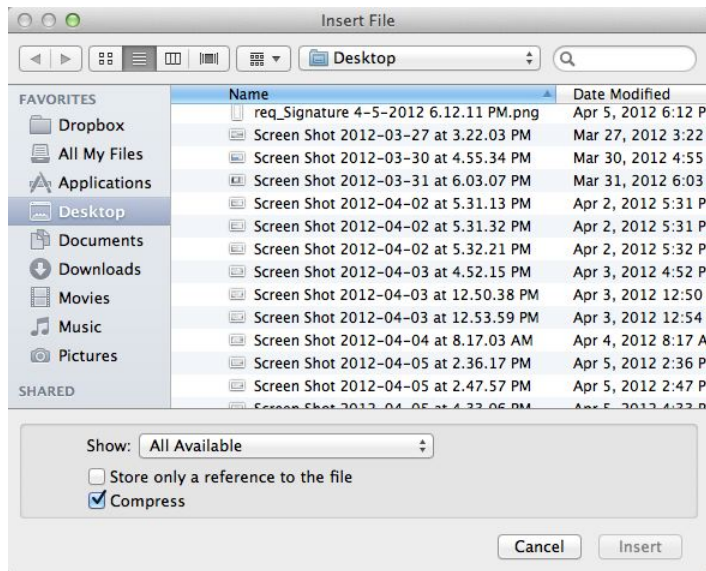


Figure 19. Use the **Compress** option (at the bottom of the **Insert File** dialog box) to compress files as they are inserted into a container field.

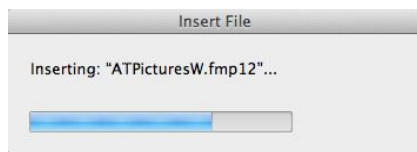


Figure 20. Progress bars appear when large files are inserted into a container field, or when a file is being compressed as it is inserted into a container field.

As already discussed, direct access to base directories should not be allowed and all external data should be managed through FileMaker Pro. Use the menu, **Edit > Export Field Contents** to move the file to another location. The file is automatically decompressed when it is exported.

Managed files stored as secure storage can be compressed. But as with their non-compressed counterparts, the files are stored as encrypted and as such can only be opened by exporting them from FileMaker Pro first. Also like their open storage counterparts, compressed secure files are automatically decompressed when they are exported from FileMaker Pro.

Note: Double-clicking a container field to open the stored file is not supported in FileMaker Pro. Double-clicking a container field to open the stored content does work on Windows if the stored content happens to be an embedded OLE object. However, Microsoft has deprecated OLE, so you should open all embedded OLE objects and save the contained document in its native format. After that you can reinsert the native document back into the container field, replacing the OLE object. See section 11.2 of the FileMaker 11 ReadMe file for more information:

http://help.filemaker.com/app/answers/detail/a_id/10007

Providing ODBC Compatibility

The CREATE TABLE statement lets you create container fields from a SQL statement using the keywords BLOB, VARBINARY, LONGVARBINARY, and BINARY VARYING. FileMaker 13 and greater provides additional keywords to create tables with externally stored and secure container fields. For example:

```
CREATE TABLE Products (Prod_Image BLOB EXTERNAL 'Files/Products_DB/' SECURE)
```

Learn more about SQL statements that support FileMaker container fields in Chapter 2 of the [FileMaker SQL Reference](#).

Using Container Functions

There are seven functions to support container fields. You can find them in the **Container** functions group in the **Specify Calculation** dialog box. Use these functions to test external file consistency, measure container data, control thumbnail generation, get metadata from files stored in container fields, and encode a file in a container field to Base64 text and decode Base64 text back into the original file format.

Base64Decode() and Base64Encode()

These two functions are used to convert container data between binary and text format known as *Base64*. Base64 is a standard for representing binary data as text. It is most commonly used as a means of transferring binary data via a channel that is capable of only transferring textual data. The binary data is *encoded* into Base64 text at the origin, and then *decoded* from Base64 text back to the original binary format at the destination. For example, you can write a calculation that uses Base64Encode() to transform an image stored in a container field into Base64 text, and embed that text within HTML generated by the same calculation to return a web page with embedded images; no external image files needed on the web server. Base64Decode() could be used if your database receives Base64 text that needs to be transformed back into the original binary format and inserted into an enhanced container field.

The Base64Encode() function takes the container field containing the file to transform as the parameter, and it returns text that is a Base64 representation of the binary file. Base64 encoding doesn't preserve the file type or name of the file, so that data will need to be provided at decode time.

Base64Decode() takes two parameters; the Base64 text, and a filename with type extension that will be applied to the resulting binary file. The binary file is returned as a container object that can then be inserted with a Set Field script step. If the filename and type aren't provided in the second parameter, then Base64Decode() returns a container object with a generic filename and extension that may not be appropriate for the content's data format.

Remember that you can use the Length() function to determine the size of container data. So getting a Length() of 0 for the result returned from either of these functions likely indicates a problem with the parameters passed in, or with the source data.

GetContainerAttribute()

Many files contain what is known as *metadata*; data embedded within the file, which describes, often in great detail, the binary content of the file itself. Examples of this are .mp3 and .m4a audio files typically used to encode music and other audio for digital playback. Audio files often contain metadata in a format known as ID3 Tags, which describe the title of the recording, the performing artist, the album name, the genre, the play length, and many other pieces of information about the recording; all of which are available without requiring the user to actually play the file.

The GetContainerAttribute() function takes two parameters; the container field to search, and the metadata attribute to find. For example, to find the album name of an audio file stored in the container field "Audio Track," you would use the following calculation: GetContainerAttribute(Music Table::Audio Track; "album").

The `GetContainerAttribute()` function lets you pull metadata from a wide variety of file formats, including the ones described above. The function may not always return attributes since audio files don't always contain ID3 Tags (or they may be stored externally to the file), image files of photos aren't always captured with location data, and so on. You can see a table of file types and potential attributes available in the `GetContainerAttribute()` section of [FileMaker Pro Help](#).

GetHeight() and GetWidth()

These two functions return the dimensions, in pixels, of the contents of a container field. They each take a single parameter—the name of the field containing the content you want to measure. If the field does not contain an image, 0 is returned.

GetThumbnail()

Normal thumbnail creation halves the original image dimensions until it's just larger than the target container field dimensions. Container data is often inserted on a layout that uses a small container object. But if you want to create a larger container object on another layout for viewing the image with greater detail, use `GetThumbnail()` to make a thumbnail that fits the larger container object.

The `GetThumbnail()` function can be used along with the `GetWidth()` and `GetHeight()` functions. It takes three parameters:

- the name of the container field that holds the original graphic file
- the thumbnail target width
- the thumbnail target height

The following formula creates a thumbnail from a container field named "imageDetail":

```
GetThumbnail (
  imageDetail ;
  GetLayoutObjectAttribute ( "imageDetail" ; "width" )
  GetLayoutObjectAttribute ( "imageDetail" ; "height" )
)
```

Or you can create a thumbnail that is a specific proportion to the original graphic with this formula. This one is half the size of the original:

```
GetThumbnail (
  imageDetail ;
  GetWidth ( imageDetail ) / 2 ;
  GetHeight ( imageDetail ) / 2
)
```

Note: You cannot use `GetThumbnail()` to change the original image's proportions. Even if the formula contains width and height data that is not proportional to the original, the thumbnail is still fitted to the original image's proportions.

VerifyContainer()

`VerifyContainer()` takes a single parameter—the container field you want to verify. It returns a Boolean value. If a file has been changed since it was inserted into the field or is missing, 0 is returned; if the file is verified a 1 is returned. Container fields with embedded data return a "?", as do empty container fields for which the, "Do not evaluate if all referenced fields are empty" option has been enabled. Use `VerifyContainer()` to enforce a consistency check before taking action that requires that the data be in the same state it was when inserted. Your script can take other action or provide feedback if the external data file is not verified.

Using GetAsText() with Container Fields

When its parameter is a container field, `GetAsText()` returns:

- The storage type, name, size, DPI and external path of an external file
- Text, if there is text in the field and it does not resolve to a valid path
- A question mark (?) for native recorded sound
- A bar code number, if the container field contains an image of type bar code. See the “FileMaker Go and Enhanced Container Fields” section below for more details.

Sample results of the function `GetAsText (toyContainer)` follows:

```
remote:puppet.jpg
size:761,708
DPI: 200
JPEG:Pictures00/toyContainer/toyID_65734/puppet.jpg
```

Each line of the result shows a label and data.

- The first line’s label shows that the field is a managed container that stores data externally. The file name follows.
- The second line shows the file size in pixels. The width is 761 and the height is 708.
- The third line shows the file’s DPI, which means Dots Per Inch. DPI is a way to measure resolution. The higher the DPI, the higher the resolution, which means the image will appear more crisp, and will often be bigger in file size.
- The forth line shows the file type and open storage path for the external data file.

Note: `GetAsText()` returns the same data as a calculation that points to a container field and has a text result. `GetAsText` is a more explicit method and is used when clarity is more important than brevity.

Controlling Container Fields

Enhanced container fields give users additional capabilities and more flexibility with the way files are displayed. However, this very flexibility may be a liability in databases where tight control over externally stored data is needed.

As a developer, your ability to control user interaction with container fields comes from different places (previously discussed in this paper):

- **field definition** - setting storage types, choosing base directories for externally stored data, and the “Maximum number of kilobytes” validation option that can limit the size of inserted files.
- **layout objects** - choosing container fields, setting optimization options, creating a script to navigate to a new layout (in converted solutions), and script triggers that can pre-flight dependent data before allowing a container field to be used.
- **calculations** – container functions, augmented with many non-container-specific functions such as using text functions to verify that the inserted file has been named in the correct format.

Using the complete suite of FileMaker Pro database, layout, scripting, and calculation tools, you’ll discover many ways of guiding user interaction with container fields to fit into any workflow.

Using the Insert File Script Step

In a robust DM/DAM system, you could create two container fields in each table that will store external data. That way you can optimize one for images and the other for interactive data. But expecting users to remember to put files into the right containers is asking for trouble.

In another situation a database that stores information about different types of design jobs might use one table for each type of data instead. Video projects would have child records in a Video table and print projects would have child records in a Print table. Whatever the database's structure, you will need to control how users insert files into container fields and which options are set for each type of file. An improved **Insert File** script step gives you control over file format filters, how a file is stored, displayed and more.

To use the options in the **Insert File** script step, move the script step into the script and highlight it to show its options:

1. Select Dialog options to view the **Insert File Options** dialog box (Figure 21).

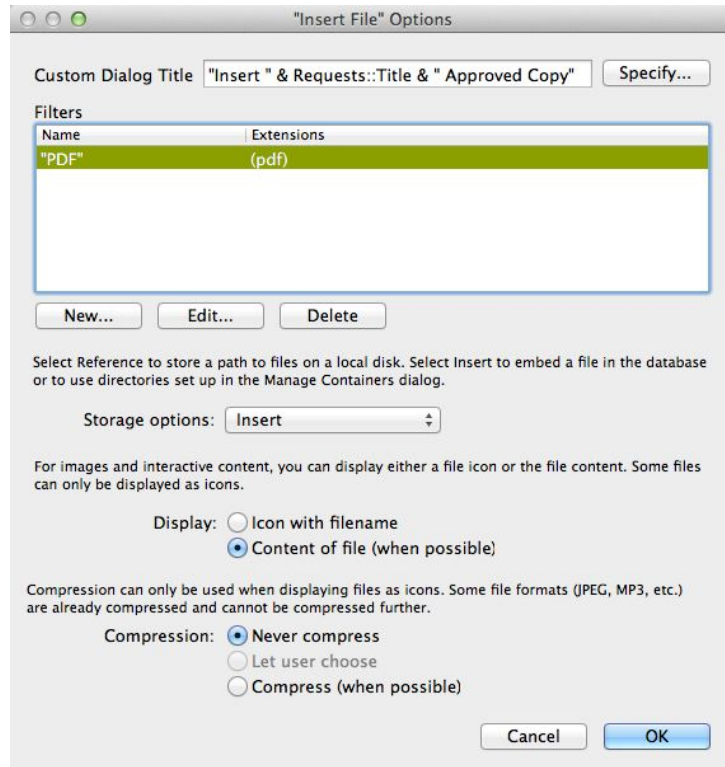


Figure 21. The **Insert File Options** dialog box lets you create custom dialog box titles, filter file formats, specify container field display and control file compression.

2. Type a custom dialog title, or click the **Specify** button to create a dynamic string.
3. Select file format filters. The default is "All files" and its extension is an asterisk (*). The user will have freedom to choose any file when the script runs. Click **New** to define multiple filters that will appear in a pop-up menu in the dialog box when the script runs (see Figure 22). Click **Edit** to change a filter or click **Delete** to remove one. However, at least one filter must be in the list, so if you try to delete the only one an error message will appear.

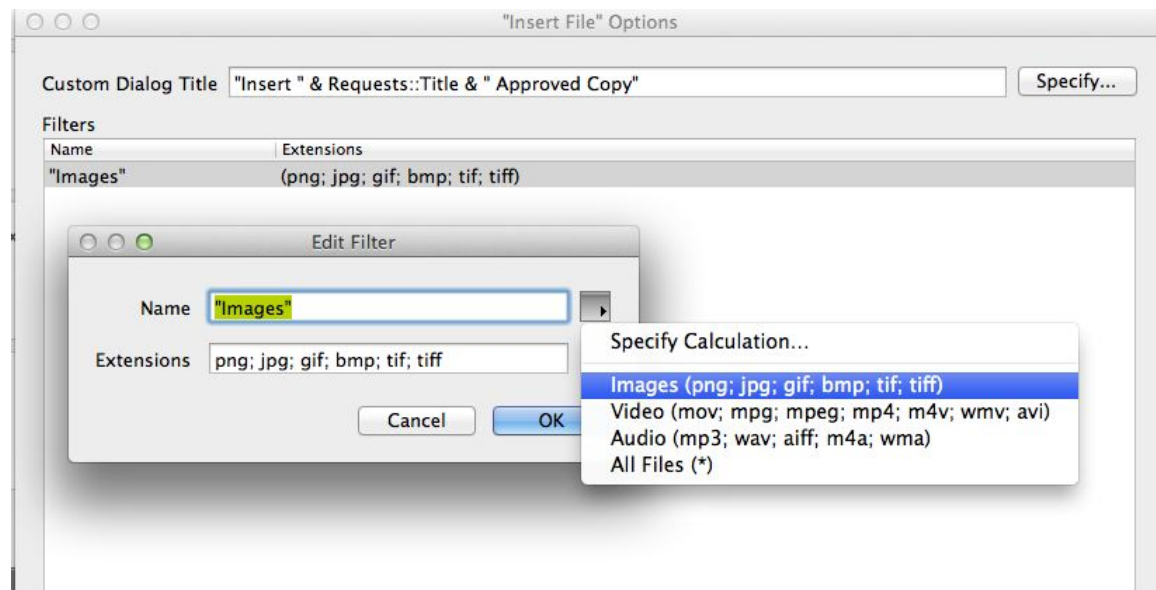


Figure 22. The **Edit Filter** dialog box lets you choose one of the pre-set options as shown here, or choose **Specify Calculation** to create a custom filter, like the one for PDFs only, shown in Figure 21.

4. Choose a storage option:
 - a. **Let user choose** gives users the same choice they would have in the native **Insert File** dialog box.
 - b. **Reference** stores a reference to a file.
 - c. **Insert** uses the container field's storage options to either embed data or store it externally.
5. Choose a display option. Keep the default option, **Content of file** (when possible), for interactive containers and images. For other file formats, choose **Icon with filename**.
6. Choose a compression option:
 - a. **Never compress** suppresses the display of this option in the dialog box when the script runs.
 - b. **Let user choose** displays the option in the dialog box. This option is only available when the display is set to **Icon with filename**.
 - c. **Compress (when possible)** requires compression of the inserted file whenever possible. Some files, JPEGs or ZIP files, are already compressed and cannot be compressed again.

Note: Compression options are not available if the storage option is set to "Reference."

The `Insert File` script step is compatible with FileMaker Pro, FileMaker Go, and runtimes. It is not compatible with FileMaker Server scheduled scripts, Custom Web Publishing, or Instant Web Publishing.

Using the Insert from URL script step

The `Insert from URL` script step downloads data using HTTP(S) and FTP(S) protocols. The step can be used to insert local files using file protocol. Targeted fields can be container or text.

When the target is a container field, supported binary formats are displayed. Non-supported file types may display as icons. Web pages with a mix of text and graphics are inserted and viewed as a file icon. When the target is a text field, data transfer is ASCII and the URL's source code is inserted.

The Insert from URL script step options are:

- Select entire contents. If this option is enabled (or if the target is a container field), all data in the target field is replaced. Although FileMaker Help states that if this option is disabled newly inserted data will be appended to the field contents, this does not apply to container fields. They hold only one file at a time and files cannot be appended in this fashion. Any file inserted into a container field with this option disabled will overwrite and replace the previous file.
- Perform without dialog. The script step's "Insert from URL" Options dialog box lets users type or paste a URL. Choose this option when the script needs to control the URL.
- Go to target field. The target field is the one you are downloading to. If no field is specified, data is downloaded into the active field—if it can accept the file's format.
- Specify URL. This option accepts a field reference, calculation or static text. Leave this choice unselected if you want users to enter a custom URL when the script runs.

The Insert from URL script step is compatible with FileMaker Pro, FileMaker Go, Custom Web Publishing, FileMaker WebDirect, and runtimes, and will also run in a FileMaker Server scheduled script.

Note: In FileMaker WebDirect, be sure to follow Insert from URL with a Commit Record or Refresh Window script step to force FileMaker WebDirect to display the new contents of the container field. Learn more about using FileMaker WebDirect in the FileMaker [WebDirect Guide](#).

Other Script Steps with Enhanced Containers

- Save a Copy As provides the "self-contained copy (single file)" option. It is available on unhosted files in FileMaker Pro.
- Open Manage Containers lets you script the opening of the dialog box. When a script containing this step has the "Run script with full access privileges" option selected, users with lower privileges can manage base directories and thumbnail generation. This script step runs only in FileMaker Pro and FileMaker Pro Advanced.
- Insert From Device is available for use only in FileMaker Go. See the "FileMaker Go and Enhanced Container Fields" section below for more.

Validating By File Size

You might want to limit inserted file size in embedded container fields to manage overall file size. You can set a maximum file size for a container field in its Validation Options panel (Figure 23).

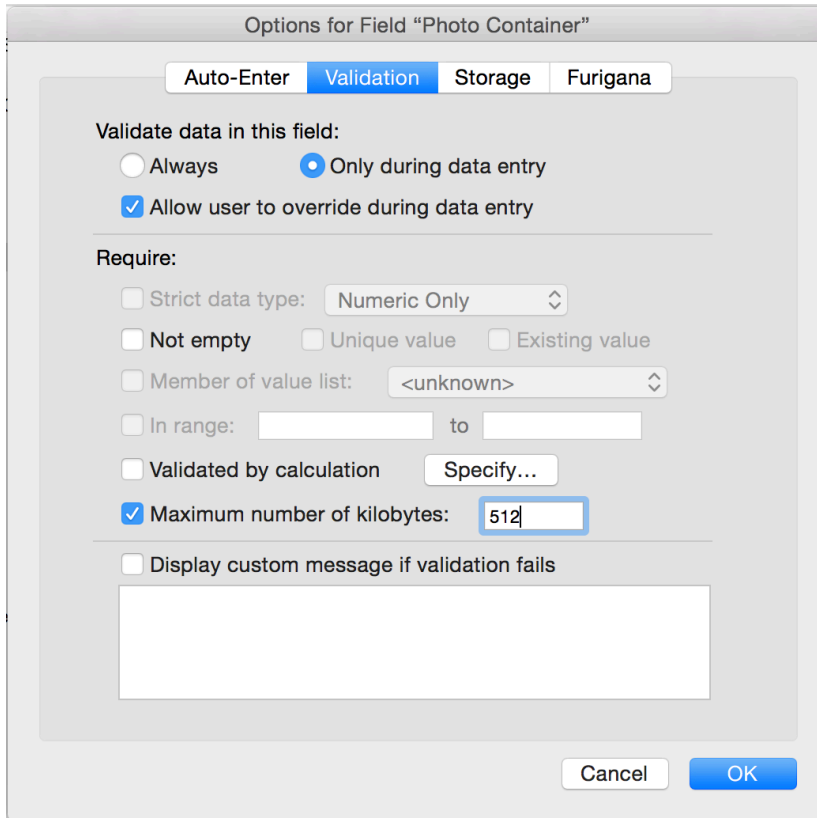


Figure 23. Select the “Maximum number of kilobytes” option and then type in your upward limit in a container field’s Validation Options. For strict control, deselect “Allow user to override during data entry” and for better user experience, provide a custom message, perhaps explaining what size limit is imposed and why.

Installing and Updating Plug-ins with Container Fields

Use the `Install Plug-in File` script step to automate installing and updating plug-ins in FileMaker Pro and FileMaker Server. The script step has a single option—the name of the container field that stores the plug-in your system needs. Plug-in files can be embedded, referenced or stored externally.

Note: Plug-ins embedded in container fields are always stored as compressed.

Plug-in files can be stored in global fields, but global containers cannot store data externally; they must be embedded. And since the data in global fields cannot be updated while the file is hosted, you will have to take your database off the server to update the plug-in file. For these reasons, global container fields are not recommended for storing plug-ins.

In addition to the stored plug-ins, you will also need supporting fields for the plug-in name and the version your solution requires. You might also want to store a brief description documenting the functions the plug-in is used for, but this is not essential to automating installations and updates. In systems that use multiple plug-ins, best practice will be to create a plug-in table, with one record for each plug-in the solution uses.

You can create a script that uses the plug-in’s external functions to compare the data you have stored to find out if the plug-in is installed, enabled and the right version. Or use `Get (InstalledFMPlugins)`, which returns display names, version number and enabled state for all installed plug-ins as a return-separated list.

Plug-ins are specific to the operating system they are run on. So if your solution is used by Mac and Windows users, you’ll need two container fields: one to store the Mac version and one to store the

Windows version of each plug-in. Then your script can use the `Get (SystemPlatform)` function to decide which plug-in version is needed for installation.

For sample scripts, including error trapping and plug-in related error codes, see [FileMaker Pro 14 Help > Designing and creating databases > Creating a database > Working with plug-ins > Plug-in update example](#).

If the installation or update fails, here are some things to check:

- All clients must use FileMaker Pro 12 or greater.
- The installed plug-in you are trying to update must be enabled. Updates can be installed for disabled plug-ins, but the updated plug-in will not be enabled.
- Plug-in preferences must have the “Allow Solutions to Install Files” option selected.
- Each plug-in must be stored in a separate container field.

The `Install Plug-in File` script step is compatible with FileMaker Pro, Custom Web Publishing, Instant Web Publishing, runtimes and can be run as a FileMaker Server scheduled script.

Plugins installed by script are stored in the following locations:

OS X

~/Library/Application Support/FileMaker/Extensions

Windows XP

C:\Documents and Settings\<user name>\Local Settings\Application Data\FileMaker\Extensions

Vista/Windows 7

C:\Users\<user name>\AppData\Local\FileMaker\Extensions

FileMaker Go and Enhanced Containers

Because of the limited disk space and memory of iOS devices, FileMaker Go on the iPhone and iPad do not have all the power of FileMaker Pro. But FileMaker Go supports managed container storage on hosted files and can use standalone databases with embedded container data.

Even though iOS devices have relatively small amounts of memory, FileMaker Go can display very large graphic files since thumbnails are displayed in enhanced container fields. Thumbnails are cached in the iOS device’s global application cache. iPads and iPhones can use “spillover” disk space if there is not enough memory available, but this on-disk cache decreases when disk space is low.

Interactive containers, including streaming audio/video and native PDF with web container toolbars, are supported. Video and audio files are displayed with the iOS Movie Player. PDF documents and URLs open in Mobile Safari.

You can tap a container field to insert data. When the container is empty, the **Import** popover menu appears (Figure 24).

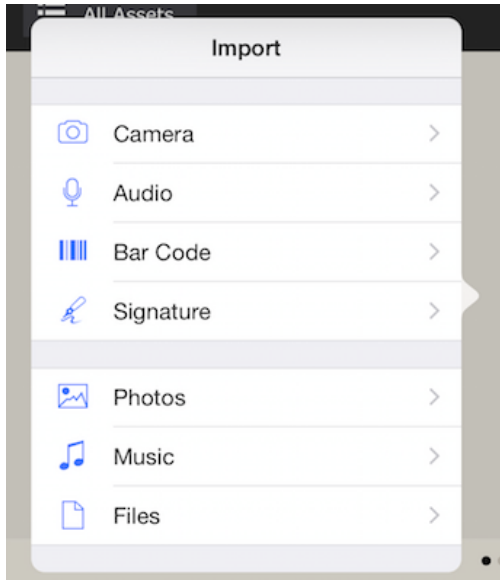


Figure 24. The FileMaker Go **Import** popover menu lets you take a photo and insert it with a single command.

- **Camera.** Take a picture or video with the device's onboard camera. A thumbnail is displayed for images. Video is displayed as an icon with filenames.
- **Audio.** Record audio with the device's microphone.
- **Bar Code.** Use the device's camera to capture a bar code. Once the barcode is detected, an image of it will be placed in the container. Use the function `GetAsText()` to extract the barcode number from a container.
- **Signature.** Capture a signature or simple drawing. Saved files are very small (typically 20-50 KB) transparent PNG files and are displayed as images.
- **Photos.** Choose a photo from the device's photo library.
- **Music.** Choose a music, audiobook or podcast file from iTunes.
- **Files.** Choose files from the Go folder that can be stored on the iOS device.

Tap the container field to enlarge images or to interact with content (Figure 25).

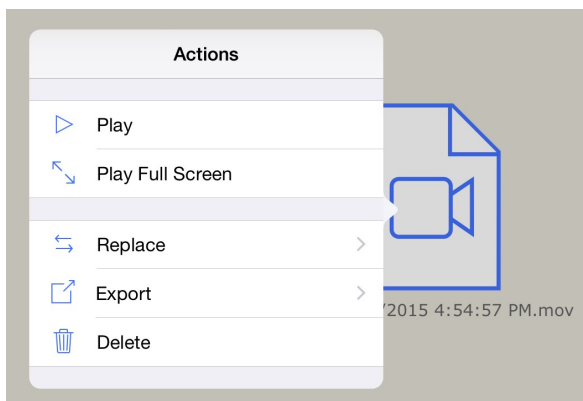


Figure 25. Tap a container field to view or play interactive content. The **Actions** popover menu appears, with the choice of View (for images, PDFs and files) or Play (for audio and video), Replace, Export or Delete. The Export command lets you email the file or save it to the FileMaker Go File Browser.

Insert from Device

This FileMaker Go-specific script step is available in FileMaker 13 and greater, and lets you create scripts that emulate the process of a user tapping a container field to insert a specific data type from the iOS device.

This script step is quite useful if you are creating a workflow that requires the user to insert a file into a container field hidden from the user; a voice memo attachment for an outgoing email message, for example. You can create an “Add Voice Memo” button on the layout that uses the **Insert from Device** script step to capture the user’s voice memo through the device microphone. The script step might look like this:

```
Insert from Device [Email::VM_File; Type: Microphone;
    Max Duration: 120; Start immediately]
```

This script step would start recording immediately through the device microphone for a maximum duration of 120 seconds, then insert the resulting sound file into the container field `Email::VM_File`, which could then be used as the file attachment on a `Send Mail` script step.

In most cases, the target field for this step will be a container field. The one exception is for bar codes, where you can specify a text field.

Note: This section is not meant to be a complete exploration of FileMaker Go. See the [FileMaker Go Development Guide](#) for more information.

AV Playback Interaction

FileMaker Go 14 provides several important enhancements for container fields containing audio or video. Playback can take place in full screen mode or inline within the layout. While the media plays, the user can continue to interact with the solution, for example entering text or clicking buttons on the layout. This new opportunity for interaction is greatly enriched though the various media-specific scripting options listed below. By combining these elements, you can come up with sophisticated solutions for digital asset review, training authentication, and other multimedia use cases.

Script steps

- AV Player Play starts the media playing.
- AVPlayer Set Options changes the playback settings for media that is already playing.
- AVPlayer Set Playback State changes the playback state of the currently playing media file.

Script triggers

- OnObjectAVPlayerChange causes a script to run when the playback state of the media within an object (i.e. container field or web viewer) changes.
- OnFileAVPlayerChange causes a script to run if a media file is already playing and the playback state is changed.
- OnExternalCommandReceived causes a script to run when the user affects playback of a media file from outside of FileMaker Go.

Function

- GetAVPlayerAttribute returns a range of attributes for the audio, video, or image file in a container field.

A simple implementation might involve a user making a comment while watching a video. When the user enters the comment field, a trigger causes playback to be paused and the current playback position to be determined. When the comment is saved, the paused position is saved with it, and the video resumes, playing automatically from that position. Because the paused position is saved, the user doesn’t need to include that information in their comment. Instead, a scripted button allows other users to view that precise moment in the video for a clear understanding of the comment’s context.

About The Author

A reformed journalist, Susan Prosser has stopped trying to bring more information to the people, and now focuses on helping them make sense of the information they already have. She is the coauthor of five books about FileMaker, most recently O'Reilly Media's *FileMaker Pro 12: The Missing Manual*. Susan is the president and owner of DBHQ (www.dbhq.net), a FileMaker consulting firm and a 2006 winner of the FileMaker Mad Dog Public Relations Award. Susan is a repeat presenter at DevCon, speaking in 2012 on Dashboard Design and Charts 101 and in 2005 on Instant Web Publishing.

© 2013-2015 FileMaker, Inc. All rights reserved. FileMaker is a trademark of FileMaker, Inc., registered in the U.S. and other countries. The file folder logo is a trademark of FileMaker, Inc. All other trademarks are the property of their respective owners. Product specifications and availability are subject to change without notice. The example companies, organization, products, domain names, e-mail addresses, logos, people, places and events depicted are purely fictitious, and any resemblance to existing persons and companies is purely coincidental. Product specifications and availability subject to change without notice. (Docv3)

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