WEB FM®

Web Server Plug-in for FileMaker® Pro 4.0

USER'S GUIDE

Web Broadcasting Corporation

555 Bryant St. #386
Palo Alto, CA 94301
USA

http://webfm.com/
sales@webfm.com
support@webfm.com

(650)329-9676

WEB BROADCASTING CORPORATION END USER LICENSE AGREEMENT

NOTICE TO END USER: CAREFULLY READ THE FOLLOWING LEGAL AGREEMENT. USE OF THE SOFTWARE PROVIDED WITH THIS AGREEMENT (THE "SOFTWARE") CONSTITUTES YOUR ACCEPTANCE OF THESE TERMS. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT, PROMPTLY RETURN THE SOFTWARE AND THE ACCORPORATIONING ITEMS (INCLUDING WRITTEN MATERIALS AND CONTAINERS) TO THE LOCATION WHERE YOU OBTAINED THEM FOR A FULL REFUND.

1. License Grant. Web Broadcasting Corporation grants to you (either as an individual or entity) a personal, non-transferable, and non-exclusive license to use the copy of the object code version of the SOFTWARE recorded on the media provided in this package. The term of this agreement and the above license will be for the duration of Web Broadcasting Corporation's copyright in the SOFTWARE; provided, that such term shall automatically expire if you breach any provision herein. Upon termination of this license, you shall return the original and all copies of the SOFTWARE and documentation to Web Broadcasting Corporation or the place where you obtained the PACKAGE, or certify to Web Broadcasting Corporation that you have destroyed the original and all copies of the SOFTWARE and documentation in your possession and within your control. You agree you will not copy the SOFTWARE except as necessary to use it on a single computer system.You agree that you may not copy the written materials accompanying the SOFTWARE. You may assign your rights under this Agreement to a third party who agrees in writing to be bound by this Agreement prior to the assignment and provided that you transfer all copies of the SOFTWARE and related documentation to the third party or destroy any copies not transferred. Except as set forth above, you may not assign your rights under this Agreement.
2. Copyright. You acknowledge that no title to the intellectual property in the SOFTWARE is transferred to you. You further acknowledge that title and full ownership rights to the SOFTWARE will remain the exclusive property of Web Broadcasting Corporation or its suppliers, and you will not acquire any rights to the SOFTWARE except as expressly set forth above. You agree that any copies of the SOFTWARE will contain the same proprietary notices which appear on and in the SOFTWARE.
3. Reverse Engineering. You agree that you will not attempt, and if you are a company, you will use your best efforts to prevent your employees and contractors from attempting, to reverse compile, modify, translate, or disassemble the SOFTWARE in whole or in part.
4. Limited Warranty. Web Broadcasting Corporation warrants that the media contained in this package is free from any physical defects and that the SOFTWARE will substantially perform in accordance with the accompanying documentation for a period of thirty (30) days from the date of purchase ("Limited Warranty"). EXCEPT FOR THE FOREGOING LIMITED WARRANTY, WEB BROADCASTING CORPORATION MAKES NO OTHER EXPRESS, IMPLIED OR STATUTORY WARRANTIES AND SPECIFICALLY DISCLAIMS ANY WARRANTIES AS TO THE PERFORMANCE OF THE SOFTWARE, NON-INFRINGEMENT OF THIRD PARTY RIGHTS, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES OR LIMITATIONS ON HOW LONG AN IMPLIED WARRANTY MAY LAST, OR THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO SUCH LIMITATIONS OR EXCLUSIONS MAY NOT APPLY TO YOU. AS TO ANY IMPLIED WARRANTY WHICH MAY NOT BE DISCLAIMED UNDER APPLICABLE LAW, THE DURATION OF SUCH WARRANTY IS 90 DAYS FROM THE DATE OF DELIVERY. THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS AND YOU MAY ALSO HAVE OTHER RIGHTS WHICH VARY FROM JURISDICTION TO JURISDICTION.

5. Customer Remedies. Web Broadcasting Corporation's entire liability and your sole and exclusive remedy under the Limited Warranty shall be, at Web Broadcasting Corporation's option, to (a) replace defective media or SOFTWARE which does not conform to the documentation or (b) authorize a refund, so long as the SOFTWARE, documentation and media are returned to Web Broadcasting Corporation with a copy of your receipt. This Limited Warranty is void if a defect has resulted from accident, abuse, or misapplication. Any replacement media will be warranted for the remainder of the original warranty period.
6. Severability. In the event of invalidity of any provision of this Agreement, the parties agree that such invalidity shall not affect the validity of the remaining portions of this Agreement.
7. No Liability for Consequential Damages. IN NO EVENT SHALL WEB BROADCASTING CORPORATION BE LIABLE TO YOU FOR ANY CONSEQUENTIAL, SPECIAL, INCIDENTAL OR INDIRECT DAMAGES OF ANY KIND ARISING OUT OF THE USE OF THE WEB BROADCASTING CORPORATION SOFTWARE, EVEN IF WEB BROADCASTING CORPORATION HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT WILL WEB BROADCASTING CORPORATION'S LIABILITY FOR ANY CLAIM, WHETHER IN CONTRACT, TORT OR ANY OTHER THEORY OF LIABILITY, EXCEED THE LICENSE FEE PAID BY YOU.
8. Export. You agree that you will not export or re-export the SOFTWARE without the appropriate United States or foreign government licenses.
9. U.S. Government Restricted Rights. If this SOFTWARE is acquired under the terms of : (i) a DoD contract: Pursuant to 48 CFR 227.7202 and its successors, use, duplication, or disclosure by the Government is subject to restrictions as set forth in this Agreement; or (ii) a Civilian agency contract: Pursuant to 48 CFR 12.212 and its successors, use reproduction, or disclosure is subject to the restrictions set forth in this Agreement.
10. Governing Law. This Agreement will be governed by the laws of the State of California as they are applied to agreements between California residents entered into and to be performed entirely within California. The United Nations Convention on Contracts for the International Sale of Goods is specifically disclaimed.
11. Entire Agreement. This is the entire agreement between you and Web Broadcasting Corporation which supersedes any prior agreement, whether written or oral, relating to the subject matter of this Agreement.

Contents

Introduction 1

About WEB•FM 4.0

WEB•FM 4.0 is a web server plug-in that enables FileMaker Pro developers to quickly and easily build wicked fast, truly dynamic and interactive online database applications for the World Wide Web or corporate Intranet. It is written as a companion to FileMaker Pro to create a web browser interface to FileMaker Pro databases.

WEB•FM does NOT rely on the built-in web services in FileMaker Pro 4.0 for publishing database information on the web. Instead, it runs as a subroutine built into Macintosh web servers providing a fast and efficient mechanism for publishing FileMaker Pro databases on the web. Just as the Web Companion is a plug-in to FileMaker Pro 4.0, WEB•FM is a plug-in to Macintosh web servers like WebSTAR, Quid Pro Quo, WebTEN, and AppleShare IP, enabling them to communicate with FileMaker Pro as a backend database application with advanced features.

WEB•FM's extensive feature list mirrors that of the FileMaker Pro application. It fully supports relational and portal fields; browsing, sorting, creating and editing database records; even running ScriptMaker™ scripts.

WEB•FM is optimized for top performance. It is fully compatible with FileMaker Pro 3.0 or 4.0, and runs native on the PowerPC™ or 68K Mac OS™ platforms. With database performance measured in ticks, not seconds, WEB•FM is capable of handling the higher traffic web sites requiring tens of thousands of database transactions per day. Of course, the faster the server, the faster the performance.

What can you use WEB•FM for? Here are just a few examples.

Example Intranet Solutions

- Corporate Directories
- Corporate Events and Calendars
- Travel and Expense Reporting
- Class Catalogs and Registrations

Example Internet Solutions

- Online Product Catalogs
- Online Order Forms
- Online Surveys
- Online Registrations

Customer Support and Registration

Please take a moment to go online, open your favorite web browser, and complete our online product registration at:

http://webfm.com/registration.html

Registration of your new software is required in order to receive technical support and announcements of software updates. Free technical phone support is available to registered users for installation and configuration support questions only. Additional technical support is available via email or through our web site.

1. Installation and configuration support questions may be sent via email to support@webfm.com.
2. "How-to" questions on implementation of specific features or solutions may be sent via email to the WEB-FM-TALK discussion list. To subscribe to this mailing list with hundreds of other WEB•FM users, please send an email message directly to requests@webfm.com. In the message body enter:

   subscribe WEB-FM-TALK

   or

   subscribe digest WEB-FM-TALK

While FileMaker Pro makes it very easy for absolute database beginners to create useful databases, the more you get to know FileMaker Pro, the more you can get out of it. Since the interaction with FileMaker Pro is happening via the web, HTML is the language used to format database information. Therefore, the more you learn about HTML, the more you'll be able to do. Please review the official HTML 4.0 specification from the World Wide Web Consortium at:

http://www.w3.org/TR/REC-html40/

WEB•FM 4.0 includes several example database solutions for you to study and learn from. In particular, the EdExpert solution includes clear annotated examples for finding records, creating new records, editing existing records, sending email, and invoking ScriptMaker scripts.

System Requirements

- Mac OS 7.5 or newer with AppleScript.
- FileMaker Pro 3.0v4 or 4.0 for Macintosh.
- WebSTAR API 1.1 compatible web server such as:
  –StarNine's WebSTAR 1.3.2, 2.1, or 3.0.
  –Social Engineering's Quid Pro Quo 2.1.
  –Tenon's WebTen 1.1.1 or 2.0
  –Apple's AppleShare IP 5.0.2

If you are not using at least FileMaker Pro 3.0v4, please download the most recent Updater from the FileMaker Inc. web site at http://www.filemaker.com/. Version 3.0v4 fixes several bugs which affect web serving and earlier versions are not fully compatible.

What's New?

WEB•FM 4.0 introduces support for easy-to-use Template files. Template files are Internet standard HTML documents void of confusing, incompatible, and proprietary markup [/tags]. For this reason a webmaster need not learn yet another markup language, and Template documents are fully compatible with all existing web page editors and client browsers.

A new highly intelligent parsing algorithm in WEB•FM automatically selects appropriate pop-up/pull-down menu items and automatically checks appropriate radio button or checkbox fields in a Template document using database record information. A context-sensitive [field name] token may be used throughout a Template document to dynamically insert cell data where desired, auto-populate pop-up/pull-down menus, and auto-replicate radio button and checkbox fields from database value lists, all with no tags required!

WEB•FM 4.0 introduces a new plug-in architecture for database-level add-on Filter and Server-Side Include Processors such as Maxum Development's NetCloak 2.5. Processors are separate web server plug-ins supporting the PIXO ("Plug-In Cross-Over") API standard for web server plug-ins to communicate and cooperate.

WEB•FM 4.0 significantly enhances features for record-level security. A fully revised FindUser command now supports most Find command features, but with record-level user authentication requirements. This means users can flexibly Find and modify their own, and only their own, records in databases where the permission to Browse records is disabled. Moreover, a client "Cookie" value may now also be used in place or absence of a valid password when finding or updating database records.

WEB•FM 4.0 includes two new bundled solutions. GOT COOKIES? is a new web server tool for easy, feature-rich "cookie" management. FM@iler is a ready-to-use program for email archival via Emailer 2.0 to a searchable web database.

General

- The fastest web server plug-in available for FileMaker Pro.
- Plug-ins for both English and Japanese FileMaker Pro.
- Improved plug-in administration database.
- Supports custom, database-level HTTP header fields.
- Configure database level Style Sheets, JavaScript, etc.
- Automatic support for custom HTTP response headers.
- Includes "Got Cookies?" for easy, feature-rich cookie management.
- Includes "FM@iler" for email archival to a searchable web database.
- Send email via Emailer 2.0.

Find Support

- Supports Find for personal, and only personal, database records.
- Supports Find on browser Cookie values.

Page Generation

- Template file support with dynamic database [field name] substitutions.
- Store Template documents inside or outside database.
- No need to learn another markup language.
- Template files fully compatible with existing web page editors.
- Automatic selection of pop-up menus, checkboxes, and radio buttons.
- Dynamic population of pop-up menus and scroll lists.
- Dynamic replication of checkboxes and radio buttons.
- Supports <portal> tag with dynamic replication of relational fields.
- Template file support with [search field] substitutions.

[Token] Support

- Token for [search field] substitutions.
- Tokens for [layout], [sort], [sortorder], [script], [html] variables.
- Tokens for [database], [field name], [page], [pages].

Security

- Find or modify personal, and only personal, database records.
- Record-level security with Find command feature set.
- Record-level security with Cookie authentication.

Extensibility

- PIXO dispatcher for add-on Processors including NetCloak 2.5.
- Processors for incoming Filters and outgoing Server-Side Includes.
- Processors for incoming/outgoing custom character translations.
- Processor for the capture and use of browser [cookie] values.

Release Notes

- Fixed multiple DoScript execution in version 3.1 on Update.
- Improved updating of portal fields.
- Replaced [next] and [prev] tokens with <next> and <prev> tags.
- Replaced Translation plug-in with Encoder Processor.
- Replaced .raw suffix flag with automatic check for custom HTTP header.
- FindUser now requires indexed Username and Password fields.

Feature List

General

- The fastest web server plug-in available for FileMaker Pro.
- Accelerated performance measured in fractions of a second.
- 68K and PowerPC native code fragment plug-in written in low level C.
- Plug-ins for both English and Japanese FileMaker Pro.
- Supports publishing up to 50 databases.
- Supports both fill-out forms and hypertext links.
- Supports interactive selection of the database task.
- Supports invoking ScriptMaker scripts in background.
- Improved plug-in administration database.
- Supports remote administration of plug-in settings.
- Supports custom, database-level HTTP header fields.
- Configure database level Style Sheets, JavaScript, etc.
- Automatic support for custom HTTP response headers.
- Includes “Got Cookies?” for easy, feature-rich cookie management.
- Includes “FM@iler” for email archival to a search-able web database.
- Send email via Emailer 2.0.

Find Support

- Supports Find on an unlimited number of database fields.
- Supports Find for either all records or just all unique records.
- Supports Find with interactive selection of search field.
- Supports Find with combined “And” and “Or” operators.
- Supports Find for a date or numeric range of records.
- Supports Find for personal, and only personal, database records.
- Supports Find on browser Username and Password values.
- Supports Find on browser Cookie values.
- Supports Find for a Random database record.
- Supports Find with “contains,” “or”, and many other operators.
- Supports Find for unique database record ID.
- Supports Find with custom header, footer, and error responses.
- Supports nested Sort of found records in ascending or descending order.
- Supports setting a maximum number of found records to return.
- Supports customizable links for <Next> and <Prev> found records.

New Record Support

- Supports Text, Number, Date, Time, Calc, and Global fields.
- Supports layout specific relational and portal fields.
- Supports repeating fields.
- Supports fields formatted as checkbox, radio button, etc..
- Supports field-level data validation.

Page Generation

- Template file support with dynamic database [field name] substitutions.
- Store Template documents inside or outside database.
- No need to learn another markup language.
- Template files fully compatible with existing web page editors.
- Automatic selection of pop-up menus, checkboxes, and radio buttons.
- Dynamic population of pop-up menus and scroll lists.
- Dynamic replication of checkboxes and radio buttons.
- Supports <portal> tag with dynamic replication of relational fields.
- Template file support with [search field] substitutions.
- Supports HTML calculation fields with access to over 100 powerful calculation functions.
- Supports the “www” field for using HTTP request values in scripts or calculations.
- Supports custom HTTP response headers for URL redirects, cookies, etc.
- Includes TAG•FM for linking databases with web page creation tools.

[Token] Support

- Supports a host of incoming HTTP request substitution [tokens].
- Supports a host of server-side include substitution [tokens].
- Supports [search field] substitution [tokens].
- Supports [variable] substitution [tokens].
- Supports the use of HTTP session [cookies].

Security

- Supports database-level, realm-based security.
- Supports database-level, administrator security.
- Supports task-level (browse, create, edit) security.
- Supports record-level Admin, Password, or Cookie edit security.
- Supports field-level security with Admin override.
- Supports record-level security with Find command feature set.
- Find or modify personal, and only personal, database records.

Extensibility

- PIXO dispatcher for add-on Processors including NetCloak 2.5.
- Processors for incoming Filters and outgoing Server-Side Includes.
- Processors for incoming/outgoing custom character translations.
- Processor for the capture and use of browser [cookie] values.

Installation 2

Installation

1 ***Drag the "WEBFM" folder into the root directory of your web server application.***

2 ***Open the "WEBFM" folder. Locate the plug-in file "WEB•FM 4.0" and drag it into your web server "Plug-ins" folder.***

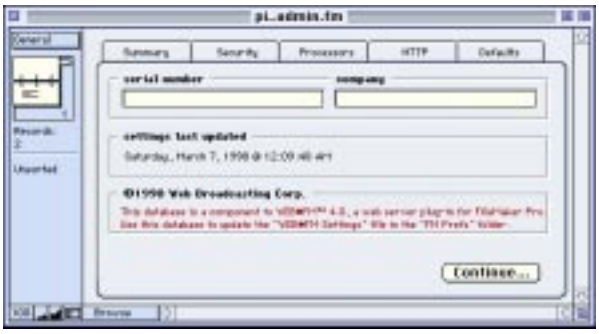
3 ***Restart your web server software.***

You should see "WEB•FM" listed in the status window as a loaded plug-in service. If a message instead says "the plug-in is out of date and cannot be used", you are using an incompatible version of the web server software and should upgrade.

IMPORTANT ***If you do not have your web server software running, the next steps cannot take place.***

4 ***Open the database file "pi_admin.fm" (Admin Database).***

It should open to a "file folder" style screen.



5 ***Enter your registration serial number in the serial number field.***

If you are evaluating WEB•FM and have not purchased it yet, you won't have a serial number.

6 ***Click the "Security" tab at the top of the screen and enter an Admin Username and Admin Password for both the "DEFAULT" database and the "PI_ADMIN.FM" database.***

N O T E ***Normally, the username and password you choose is the same username and password you chose for the web server "pi_admin" security realm.***

7a *For PowerPC Web Servers only:*

After you have entered your chosen Admin Username and Admin Password, click the Update button. In the dialog that appears (pictured below), choose your web server application and click OK.

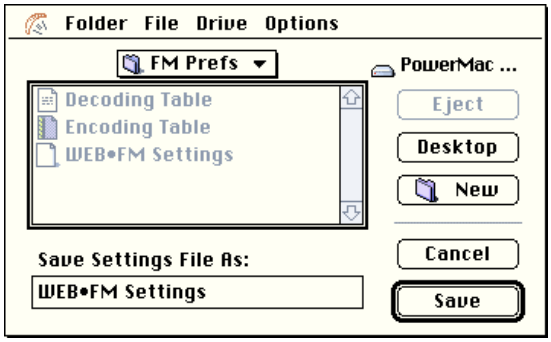
WEB•FM will then update the settings file in the "FM Prefs" folder with new preferences. The web server should display a message "The settings have been updated successfully".

N O T E ***If the dialog pictured below does not appear, make sure that AppleScript is installed and enabled, and that the scripting addition "Choose Application" exists in the system's Scripting Additions folder.***

7b *For 68K Web Servers only:*

After you have entered your chosen Admin Username and Admin Password, select the "Export..." command from the Script menu.

In the dialog that appears (pictured below), save the "WEB•FM Settings" file in the "FM Prefs" folder in the Plug-ins folder, replacing the previous settings file. Restart your web server software.



8 ***Close the "pi_admin.fm" database file.***

This database may remain open and is web enabled, for remote administration, but for extra security make sure a security realm for "pi_admin" exists.

Verifying Installation

When installation of WEB•FM is complete, it's a good idea to make sure configuration was successful. To test WEB•FM you *must* access documents through your web server (http://), *not* directly from the hard drive (file:///). This means you *must have your machine acting as a web server*, and you must enter in your web browser a URL location beginning with "http://".

TIP ***If this is your first time dealing with web server software, read the side bar "Testing Your Forms."***

To test installation:

1. ***Make sure your machine is running as a server using WebSTAR or another W*API 1.1 compatible Mac OS web server.***
2. ***In the "EdExpert" folder, open the "EdExpert.fm" database with FileMaker Pro.***
3. ***Using a web browser (such as Netscape 4.0) access the "EdExpert" example on your machine by entering a URL location similar to:***

   http://your.domain.com/webfm/edexpert/

4 ***Make a selection from one of the pop-up/pull-down menus and submit the search form.***

A “hit list” of found records should return with links to more detailed information. The FORM contents submitted by the web browser and received by the web server were used by WEB•FM to query the “EdExpert.fm” database. WEB•FM returned the found set of records as HTML.

T I P ***If large amounts of garbled text and HTML are returned to the web browser, this means WEB•FM did not load properly and the web server returned the raw database file directly from disk.***

TESTING YOUR FORMS

After installation and configuration of WEB•FM, it's wise to make sure installation was successful. To test WEB•FM you'll need to access your FORMs through your web server.

Since you also need to have the server software running to complete the installation, this is a good time to read this and follow these instructions.

The most convenient way to test your FORMs is to have your own machine act as a server.

One way to do this is to simply launch your server software. On a corporate LAN this is normally very easy. If you use a modem let it connect to your ISP. However, if you are going to spend hours at a time developing and testing, you will tie up your ISP connection long past what is reasonable.

The better way to test is to fake your machine into serving itself. An excellent source of information on how to do this is provided by StarNine, authors of WebSTAR, at <http://www.starnine.com>. Under support, they have a fact sheet entitled, "WebSTAR: Pre-Installation Issues."

Here are the basics:

- *Open the TCP/IP control panel. Go to Configurations, duplicate your configuration, renaming it accordingly and change "Connect" from whatever it is to "AppleTalk (MacIP).*
- *"Configure" should change to "Using MacIP Manually".*
- *Assign an IP Address of 192.168.0.2 and close TCP/IP.*

Now when you start your server software it won't try to dial. It will simply accept the IP address as the one you entered. You'll see your IP number selected.

If you are not running Open Transport, do the same type of thing in the TCP control panel you have. If you are on Ethernet or such refer to the web server docs or experiment.

Getting Started 3

Working with FileMaker Pro

To make good use of WEB•FM you need to be very familiar with FileMaker Pro. It is beyond the scope of this manual to serve as a tutorial for FileMaker Pro, but there are some basic characteristics and functions that you need to be aware of. First, let's deal with some basics that you need to know in order to make WEB•FM really work for you.

Also, review the official HTML 4.0 specification from the World Wide Web Consortium at: http://www.w3.org/TR/REC-html40/

Naming Hints

Although not required, single word database file names and field names may make your work a little easier. If you are creating a database for the first time for use with WEB•FM, keep field names short and, if possible, avoid spaces. Name your databases succinctly, and stick to the standard alphabet. Avoid using odd characters such as "/","#","< >", etc..

Another excellent idea is to append ".fm" to the end of the name of your database file. It gives you an extra margin of security (explained in the section on security). However, it's not absolutely necessary, so don't worry if you already have your database file names and relationships all set up.

Reserved Field Names

WEB•FM is designed to call upon certain database fields with specific names. Reserved fields are fields often used or required by WEB•FM which allow you to deliver formatted HTML as a response to a database request. You will find more detail on what to put into these fields when you get to the section on Page Generation. For now, just be aware that you should keep them in **reserve** for use with WEB•FM.

html The html field generally contains basic "hit list" information for each individual record in a found set that is returned by WEB•FM to the web browser. Typically, it also provides a link to the detail field or Template file which enables a user to select one specific record from the found set for retrieving more detailed information.

detail The detail field is sometimes used in place of a Template file to deliver detailed information specific to a record you want to return to a web browser. Detail is created as a text calculation field that concatenates (links together) the information from other fields in a specific record and prepares them for delivery in formatted HTML.

header The header field is typically created as a global field in FileMaker Pro. WEB•FM automatically gets header HTML from the header field whenever the Find or FindAll commands are used. If you leave this field out of your database, a default header defined in the Admin database is supplied.

footer The footer field works in exactly the same manner as the header, except it's placed at the bottom of the page.

error The error field allows you to return a customized error message to the web browser whenever a database task can not be completed (e.g. no records are found that match the search criteria). WEB•FM will look for this field whenever an error occurs in the search process. If you leave this field out of your database, WEB•FM will supply a default error.

www If formatted as a global repeating field with 20 repetitions, the www field will temporarily capture many of the client HTTP request parameters received by the web server. These parameters or "environment variables" can serve various purposes. For more information see the section on Advanced Features.

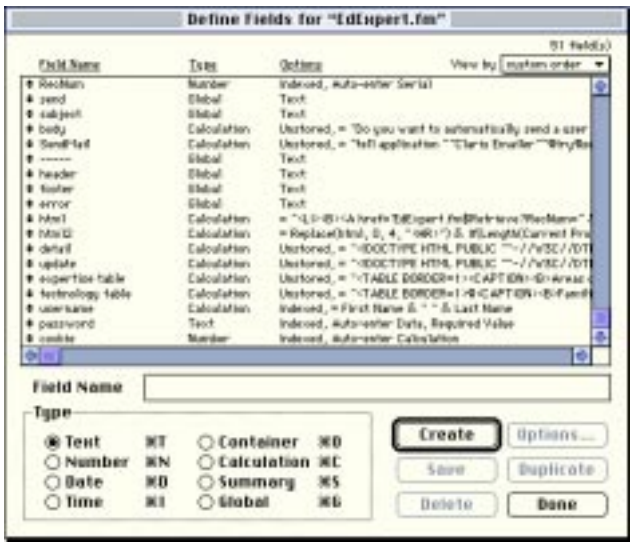
username, password The username and password fields are required for editing or deleting an existing record. They should normally be indexed fields of type text. These fields enable every record to have a different username and password for record-level security.

cookie The cookie field is equivalent to, and may be used in place of, the username and password fields. This field enables authorization to edit or delete an existing record using a browser HTTP cookie value.

recid The RecID field is actually an automatic, internal database field not visible in Define Fields. Never create a database field with this name. WEB•FM will sometimes find a specific record using a RecID value.

recnum Every database should have an indexed unique serial number field, normally of type number. This field is useful to WEB•FM in searching for and identifying individual database records. If you are starting your database from scratch, create this field with the option to auto-enter a serial number. If you are modifying an existing database, you need to create the serial number field, Find All records in your database, and use FileMaker Pro's Replace command (found in the Edit menu) to enter a new set of serial numbers.

Now that you are aware of reserved field names, let's look at what you do with some of those fields.



Publishing a Database File

WEB•FM 4.0 relies partly on Template files and partly on database calculation fields to dynamically build and deliver HTML documents for publication on the web. Template files are easier to use and design, but HTML calculation fields can be far more powerful. In the case of calculation fields, WEB•FM's job is to grab HTML from these fields and serve them to the web browser. By relying on calculation fields you have complete control over the HTML formatting of your documents, including direct access to over 100 powerful calculation functions supported by FileMaker Pro.

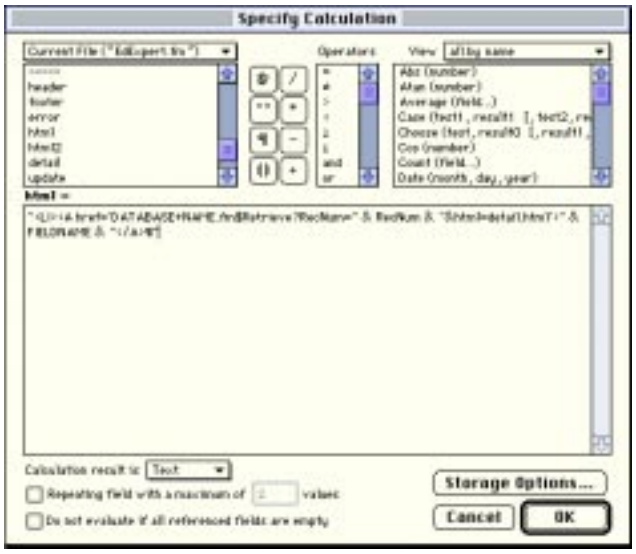
The following few pages go through the precise process of writing an HTML page inside a FileMaker Pro calculation field. We will take a look at the required "html" calculation field, and the optional "detail" calculation field. Keep in mind that only the "html" calculation field is required for the purpose of returning a found set of records. Most of your dynamic web pages that return a single record of information will be Template files as discussed in Chapter 7: Page Generation.

1 ***Make a copy of an existing database file you have already, or make a new database file altogether with fields for a solution you wish to publish on the web.***

2 ***Go to "Define Fields..." and create a new field of type calculation with name "html".***

The "html" field is the most important field you will add in order to web-enable your database.

3 ***In the "Specify Calculation" dialog for this field enter the calculation formula in the screen shot shown below.***



4 ***Replace "DATABASE+NAME" with the actual name of your database. Any spaces in the database name should be replaced with a "+" character.***

5 ***Replace "FIELDNAME" with an actual field name from your database.***

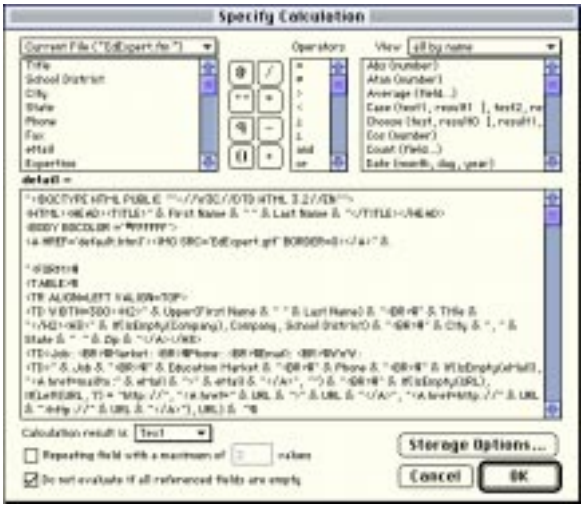
6 ***If you have a unique serial number field with a name other than RecNum, then replace "RecNum" with the name of your unique serial number field.***

7 ***Select "TEXT" as the calculation result and click OK.***

N O T E ***Steps 8 and 9 below are optional. They are presented here only as an example of a complex detail calculation. To return a "detail" field, the "html" calculation formula shown above should be modified, replacing "html=detail.html" with "html=detail".***

8 ***Go to "Define Fields..." and create a new field of type calculation with name "detail".***

9 ***In the Specify Calculation dialog for this field enter a simple calculation formula for evaluating a detailed HTML document for a specific database record. The screen shot below is a complex example from the EdExpert.fm database.***



10 ***Click the Storage Options button and check the option for "Do not store calculation results — calculate only when needed".***

11 ***Exit "Define Fields...".***

ABOUT QUOTATION MARKS

One naggy little problem with the creation of calculation fields is that proper HTML calls for quotation marks surrounding a URL location or entity attribute.

FileMaker Pro interprets a double quote surrounding a URL location or entity attribute as the beginning or end of a text string and hence will not generate the double quote in the resulting HTML text.

Fortunately, there are other options for including quotation marks in a calculation formula. The easiest is to use a ***single*** *quote mark anywhere a quote mark in a text string is needed. The HTML specification allows for either single or double quotes surrounding a URL location or entity attribute. This does not apply to what's known as"smart" or "curly quotes", so go into your FileMaker Pro document preferences and turn "curly quotes"off.*

Additional Hints

Calculation fields can increase the size of a database very quickly. Also, editing a calculation field again and again to get proper HTML can be time consuming when the database is large and FileMaker Pro performs a recalculation every time a modification is made. To make the task of creating and tweaking your database fields less painful, it's a good idea to set the "Storage Options" for the field's calculation to "Do not store calculation results — calculate only when needed".

However, this recommendation may not apply to the "html" field containing only a small amount of information that is typically returned in a "hit list". Certainly, indexing of HTML calculation fields can and should always be turned off.

Returning a Custom Header and Footer

WEB•FM provides a default setting utilizing the variables Header and Footer which allow you to return a customized header or footer from global fields that you set up in the database.

Use “Define Fields…” in FileMaker Pro and add new fields to the target database of type Global with the name “header” and “footer”. In Browse mode paste the HTML for a custom header into this “header” field and custom footer HTML into the database “footer” field. Now, whenever a database query occurs with the Find or FindAll commands, the custom HTML in these two fields will be returned to the web browser as portions of the HTML document. The “header” and “footer” HTML can be returned when other commands are used, but in such a case it must be explicitly specified in a FORM or a Link by using the INPUT variables “header” and “footer”.

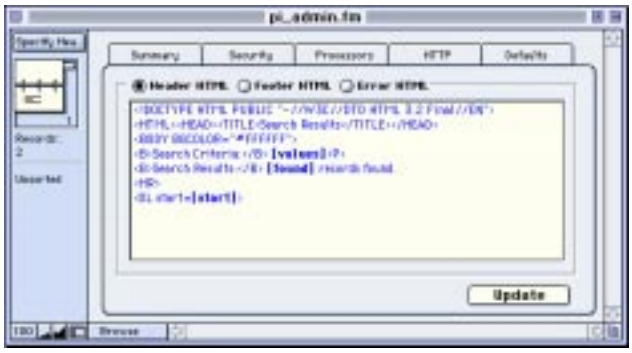
Example:

```
<INPUT TYPE=hidden NAME=header
VALUE=header>

<INPUT TYPE=hidden NAME=footer
VALUE=footer>
```

Returning an Error When No Records Are Found

Global fields in FileMaker Pro allow you to create a cell of information that is the same in every record. With regard to WEB•FM, Global fields work very well as HTML delivery systems for elements that will be the same on any page (header and footer fields, for example), as well as self-contained messages that you want delivered when something is amiss (error messages would be the most common example of this type of use).

Use “Define Fields…” in FileMaker Pro and add a new field to the target database of type Global with the name “error”. In Browse mode paste the HTML for a custom error message into this database “error” field. Now, when a Find results in no records found, the HTML in the “error” field will be returned to the web browser by default.



Basic Commands

4

WEB•FM supports eight commands for acting on a FileMaker Pro database. These commands closely mirror the tasks available in FileMaker Pro. INPUT variables allow further control over command behavior, influencing how WEB•FM interfaces with a target database. A client browser submits the contents of a FORM (or Link) along with a command and variables to an origin web server. The server passes the request to WEB•FM which acts accordingly on the target database based on the command and variables present.

Commands are normally specified in a FORM using the NAME attribute of an INPUT element as shown here:

```
<INPUT TYPE="submit" NAME="Find"
VALUE="Submit">
```

Commands may alternatively be specified as part of a FORM action following the database name and "$" character.

```
<FORM METHOD="POST"
ACTION="Database+Name.fm$Find">
```

In the case of a hypertext link, commands may be specified after the database name and "$" character, or in the name/value string portion of the request URL.

```
<A href="database+name.fm$Find?sort=date">
```

or

```
<A href="database+name.fm?sort=date&find=">
```

If you omit the NAME attribute from the submit button, or for whatever reason a command is not specified or received by WEB•FM, then by default the command is assumed to be Find. You can change this default command setting globally by using the Admin database and selecting an alternate default command.

Command syntax for using a graphic as a Submit button on a form is as follows:

```
<INPUT TYPE="image" NAME="Find"
SRC="submit.gif">
```

Find Find is equivalent to selecting Find from the Mode menu in FileMaker Pro and performing a Find for entered query values. There is no limit on the number of fields which can be searched. Find returns a list of found records to the web browser matching the search criteria with a header and footer, or an error page if no records are found.

FindUser FindUser is a very powerful command. There is no equivalent in FileMaker Pro. It is perhaps analogous to performing a Find in FileMaker Pro based on the System setting for current User Name. It supports most Find command features, but with record-level user authentication requirements using the username and password values (or HTTP cookie value with name "cookie") from the client browser. This means users can flexibly Find and modify their own, and optionally only their own, database records. A cookie value may be used in place or absence of a valid password when finding or updating database records.

This command requires an indexed "username" and "password", or "cookie" database field. It can be used very nicely to retrieve an update form when the user knows the correct username and password for a specific record. It is most useful with ecommerce and membership directories where each member needs to find and modify their own individual record information.

FindAll This command works exactly the same way as the Find All command in FileMaker Pro, finding every record in the target database. FindAll will optionally return a database value list containing only unique database records.

Retrieve This command is similar to Find. It differs from Find in that by default, no “header” or “footer” is returned. Where Find is meant to return a group or “hit list” of records from a search FORM, Retrieve is typically used in a Link to get a single specific detail record of information via a Template file.

Add Add is equivalent to selecting New Record from the Mode menu in FileMaker Pro and entering new values or data into field cells. Within FileMaker Pro, the user can select from pop-up menus, pop-up lists, check boxes or radio buttons. The same is true with a web browser interface via WEB•FM. Just as it doesn’t matter which order you enter information into the fields inside FileMaker Pro, it doesn’t matter in which order you enter information from the web form.

Update This command edits a single found record with submitted FORM data. Similar to the FindUser command, it uses the submitted Field/Value variables (if any), or the username and password values (or “cookie”), in an attempt to find the correct record to update. It then compares the browser username and password (or “cookie” value) against the username and password database fields (or “cookie” field) in order to validate authorization to perform an update. If more than one record is found, or no exact matches are found, or the password supplied is not valid, then access is denied and the user is prompted to supply a valid password before trying again.

Delete Delete is equivalent to selecting Delete Record from the Mode menu in FileMaker Pro. It works the same way as Update. However, it will delete the found record rather then edit it, and the HTML page is built prior to the record being deleted.

Random This command has no direct equivalent in FileMaker Pro. It's default behavior is to Find a random record. The Field and Value variables may be used to find a random record from those matching some specific criteria.

As you can see, the same basic tasks you use within a normal FileMaker Pro database — browse records, create records and edit records — are available with a web browser interface by using WEB•FM.

INPUT Variables

5

WEB•FM supports a variety of user-selectable variables for customizing the behavior of a database transaction. Some of these variables are required when using a specific WEB•FM command. Other variables are optional or may have a default value when one is not submitted as specified in the Admin database.

Variables may be specified on either a fill-out FORM or in the URL of a hypertext link. When in a FORM, the variable field may be hidden or specified as a checkbox, radio button, or select menu. When specified in the URL of a hypertext link, variables should go after the “?” character in the URL as a string of name=value field pairs with “&” delimiters.

Header Header specifies a database field to return with header HTML displayed above a found record or list of found records. The header variable may be used with any command. If you are using the Find or FindAll command, and you have placed a “header” field in your database, WEB•FM automatically returns the “header” database field. If no “header” field exists, then the global header HTML in the Admin database is returned.

Example:

```
<INPUT TYPE=”hidden” NAME=”header”
VALUE=”header”>
```

Footer Footer specifies a database field with HTML to return at the bottom of a found record or list of found records. May be used with any command. If you are using the Find or FindAll command, WEB•FM automatically returns the “footer” database field.

Example:

```
<INPUT TYPE=”hidden” NAME=”footer”
VALUE=”footer”>
```

Error Error specifies a Template file or database field to return when no records are found which match the search criteria, or there is some other problem performing a database transaction. May be used with any command. When not specified, WEB•FM automatically returns the “error” database field by default.

Example:

```
<INPUT TYPE=”hidden” NAME=”error”
VALUE=”error.html”>
```

HTML HTML specifies a Template file or database field to return as the main body in a reply document. If omitted, by default WEB•FM returns the “html” database field. The VALUE of the html variable is

equal to the requested Template file name in the same relative directory for the current request, or the desired database field name. Template files must end with a ".html" suffix, while database field names normally do not include a ".html" suffix but can if Template substitutions are needed. The ".html" suffix is a flag to WEB•FM to perform database field and value list substitutions as described in Chapter 7: Page Generation.

Example:

```
<A
href='EdExpert.fm$Retrieve?RecNum=1&html=
detail.html&layout=data+entry'>
```

Operator Operator specifies how a database Find should "operate". If omitted from a form, the default "begins with" operator is assumed. Operator is NOT field specific but applies to all fields in the entire form. However, including comparison operators within a field's search criteria can sometimes override the global operator. In particular, the "request" option for the Operator variable allows find symbols such as "...", "=", "//", and "!" to be recognizes by FileMaker Pro. Both comparison and logical operators are available as follows:

begins with	less than
contains	less than or equal
ends with	not equal to
equals	request
greater than	or
greater than or equal	

Example:

```
<A href='EdExpert.fm?date=//...3/9/
98&operator=request'>
```

Field Field specifies a database field name to search on. This variable is always partnered with the Value variable which tells WEB•FM what to search for. Although normally used with the Find command so the desired search field may be user selectable, Field/Value may be used or even required with most commands in order to perform special needs. It is highly recommended for use with the Update and Delete commands to specify which record should be edited.

Example:

```
<SELECT NAME="field">

<OPTION>City

<OPTION>State

<OPTION>Country

</SELECT>
```

Value Value specifies a query value to search for in the database field specified by the Field variable. Field and Value are always used together, most of the time in an update FORM.

Example:

```
<FORM METHOD="POST"
ACTION="database+name.fm">

<INPUT TYPE="hidden" NAME="field"
VALUE="RecNum">

<INPUT TYPE="hidden" NAME="value"
VALUE="1234">
```

Sort Sort specifies a database field for sorting the found records. Two Sort variables may be used for nested sorts in a single form or Link.

Example:

```
<INPUT TYPE="hidden" NAME="sort"
VALUE="date">
```

SortOrder Specifies the direction for sorting found records. SortOrder requires the Sort variable in order to specify which database fields are to be used for sorting. SortOrder recognizes “ascending” and “descending” as valid options. However, “ascending” is WEB•FM’s default behavior, so the variable is only necessary if you want to sort the records in descending order.

Example:

```
<INPUT TYPE=”hidden” NAME=”sortorder”
VALUE=”descending”>
```

Max Max allows you to limit the number of found records returned to the Web browser. The value of Max must be an integer. If this variable is omitted WEB•FM will return all found records to the client browser. For larger databases it’s reasonable to disable Browse All permission in the Admin database.

Example:

```
<INPUT TYPE=”hidden” NAME=”max”
VALUE=”10">
```

Layout Specifies a database layout name to use when performing a database request. Layout is required when dealing with ANY relational fields in any manner, and is highly recommended for use with Template files and all commands save Find or FindAll.

Example:

```
<A
href=’EdExpert.fm$Random?html=detail.html
&layout=data+entry’>
```

T I P ***By creating a database layout which includes only the fields you wish to use on the Web, and then targeting that layout by using the Layout variable, you can often improve the speed of a database request.***

DoScript Use DoScript to invoke ScriptMaker scripts you have created in your FileMaker Pro database. DoScript tells WEB•FM the name of a ScriptMaker script to run and will invoke that ScriptMaker script in your database prior to returning html. Scripts should normally be visible in the Script menu, so select the "Include in menu" script option. The point at which WEB•FM invokes a script is as follows:

1. Find, Edit, or Add records
2. Invoke a ScriptMaker script
3. Sort found records
4. Return html

Example:

```
<INPUT TYPE="hidden" NAME="doscript"
VALUE="sendmail">
```

FORM Basics

6

Most often, you will access information in your database through an HTML fill-out form. Mandatory attributes for the FORM element include METHOD and ACTION, where METHOD specifies how the form contents are sent to the web server, and ACTION is the URL of the web server to which the form contents will be submitted.

WEB•FM uses the ACTION attribute in the FORM element to specify the name of the target FileMaker Pro database to receive the form contents. Including a ".fm" suffix on the end of the database name in the URL will flag the web server in recognizing the request as a database transaction to be handled by WEB•FM. Although recommended that the actual database file name optionally have a ".fm" suffix, it's not required. Still, if you store the database in the web server directory tree, then it is a good idea to name the database with a ".fm" suffix for security reasons.

Because space characters in a URL are illegal, and many database documents include spaces in their name, it's important that we replace these space characters, if any, in the URL with a "+" character.

Example:

```
<FORM METHOD="POST"
ACTION="database+name.fm">
```

In a form you can include INPUT, SELECT, and TEXTAREA tags to specify fields and their interface elements. As you create a form, your primary task is to decide which database fields you want WEB•FM to handle, then place INPUT elements in your form that match the names of those database fields.

IMPORTANT NOTE

The field names in the database that you want to Find on or Add to must match EXACTLY the INPUT elements on your FORM. If the INPUT elements on a form do not match the database field names EXACTLY (spelling and punctuation) WEB•FM will not be able to map values to database fields.

An INPUT element of type "submit" is a pushbutton that causes the current form contents to be packaged into a string of "name=value" pairs and sent to an origin web server. WEB•FM expects the name (not the value) of the submit pushbutton to match the command desired for the form. When the name of the submit button is not recognized as a valid command, WEB•FM will default to performing a Find request using the submitted form values. Legal commands are:

Find	Random
FindAll	Add
FindUser	Update
Retrieve	Delete

TIP ***"A 'submit' element is required in all forms except those containing only a single INPUT element of type TEXT (in which case Return in the text entry area submits the form) or at least one INPUT element of type IMAGE (in which case a click in the image submits the form)."***

Building a Find Form

Begin your form with standard DOCTYPE, HTML, HEAD, and BODY elements and follow with a FORM element that specifies the target database.

Example:

```
<FORM METHOD="POST" ACTION="EdExpert.fm">
... </FORM>
```

Next, place inside the form variables for returning the contents of the 'header', 'footer', 'html' database fields to the web.

```
<INPUT TYPE="hidden" NAME="header"
VALUE="header">

<INPUT TYPE="hidden" NAME="footer"
VALUE="footer">

<INPUT TYPE="hidden" NAME="html"
VALUE="html">
```

Next, place inside the form variables for sorting all found records, while returning only a few.

```
<INPUT TYPE="hidden" NAME="sort"
VALUE="last name">

<INPUT TYPE="hidden" NAME="max"
VALUE="10">
```

Next, place inside the form INPUT elements for fields you wish to search on. The following HTML allows a user to search the ExExpert.fm example database by username and company.

Example:

```
Name: <INPUT NAME="username" VALUE="">

Company: <INPUT NAME="company" VALUE="">
```

Specify the Find command in the NAME attribute of the 'submit' pushbutton.

Example:

```
<INPUT TYPE="submit" NAME="Find"
VALUE="Submit">
```

The form you just created allows a user to search by name and company. It then sorts all found records, returning HTML from a maximum of 10 records.

Finally, in case no records are found we need to include the Error variable with a value for an error Template file located on disk in the same relative directory as the current URL.

```
<INPUT TYPE="hidden" NAME="error"
VALUE="error.html">
```

Finding on Relational Fields

To search on a relational database field, include the variable Layout in your form with a value that exactly matches the name of the database layout containing the relational field. Relational and portal fields must exist on the target database layout in order to use them.

Relational fields must be referred to by a special name that begins with the name of the database relationship, followed by two colons "::" and the actual field name.

Example:

```
<INPUT TYPE="hidden" NAME="layout"
VALUE="layout name">
```

```
<INPUT NAME="relationship::field name"
VALUE="">
```

N O T E **The relationship name is NOT the name of the related database. It is the name you gave the relationship in the "Define Relationships..." dialog. (For more information, refer to the FileMaker Pro documentation on relational fields).**

Using Operators

A typical form will query a target database with an "AND" find criteria. In other words, every query term submitted must be present on a record for that record to be found. To change the method in which records are searched, include the variable Operator in a form with a value for the type of search desired. As mentioned previously, there are 11 options for Operator that are largely self explanatory. These options apply to the entire form and are NOT field specific. "Begins With" is the default query method for a find unless otherwise specified.

FileMaker Pro specific symbols for finding duplicates, current date, single character or invalid date or time are NOT by default recognized characters. If a search requiring one of these special characters in the query text is required, you MUST use the "request" value with the Operator variable.

The "OR" operator tells WEB•FM to search ALL submitted fields with an OR criteria, with one exception. The Field and Value variables always use the AND operator. Essentially, this provides a combined AND/OR find criteria.

Massaging Found Records

Sorting Found Records

Sort is an INPUT variable whose value is a field name in the target database. When a form is submitted, WEB•FM intercepts this variable and sorts the found records by the field name specified. Nested sorts are possible by simply having two interations of the Sort variable, each having a different value. You can make the Sort variable a select menu or another type so the user can choose which database field to sort on.

Example:

```
<INPUT TYPE="hidden" NAME="sort"
VALUE="city">
```

T I P ***For performance reasons, WEB•FM does not always show a found set of records in the database unless a sort or script is required. To force showing all records found without actually sorting, include the Sort variable but without a field name value.***

Sorting found records in ascending or descending order is possible by including the variable Sortorder in a form with a value of “ascending” or “descending”. As mentioned before, since WEB•FM sorts records in ascending order by default, it is only necessary to use this variable if you need to sort in “descending” order.

```
<INPUT TYPE=”hidden” NAME=”sortorder”
VALUE=”descending”>
```

T I P ***Refer to Chapter 8: Processors for information on using a Filter processor to filter incoming requests using an alternate to the sortorder variable of “sort=up” or “sort=down”.***

Returning Found Records

Include the variable HTML in a form with a value equal to a database field name containing the HTML to be returned after a successful Find. Should this variable be omitted, the default is to return the database field named “html”. Refer to the EdExpert solution for an example of using the HTML variable in a form as a radio button so the user can dynamically choose the database field to return - and hence the preferred HTML formatting for the “hit list” of found records.

Example:

```
<INPUT TYPE=”hidden” NAME=”html”
VALUE=”html”>
```

After a successful Find, the contents of the field "html" are returned by WEB•FM from each record found. After submitting the search form the web browser will display what is commonly referred to as a "hit list" of found records. The HTML returned to the web browser is the result of FileMaker Pro evaluating the field's calculation formula for each record. The formula includes normal text, HTML tags, and most importantly the values from other database fields who's information is desired. This is accomplished by concatenating fields in the formula with HTML.

The simplest formula for a basic "html" field utilizes the Retrieve command and might look something like this example:

```
"<LI><A
href='EdExpert.fm$retrieve?RecNum=" &
RecNum & "&html=detail.html'>" & Company &
"</A><P>"
```

Returning a Range of Found Records

It's often better to return only a range of found records rather then all records found with a successful find. Use the variable Max in your form with a numeric integer value equal to the maximum number of records you wish to return with a single database Find.

Example:

```
<INPUT TYPE="hidden" NAME="max"
VALUE="10">
```

Refer to the section on [token] support for information on including a "Next Page" link for displaying the next or previous [max] set of found records.

Creating a New Record

Begin a new record the same way you started the previous FORM; by creating a URL that tells WEB•FM what database to work with.

Example:

```
<FORM METHOD="POST"
ACTION="Database+Name.fm">
```

Next, include INPUT elements that match exactly the field names in the FileMaker Pro database that you are working with.

Example:

```
Name: <INPUT NAME="first name"
VALUE=""><P>

Company: <INPUT NAME="company"
VALUE=""><P>
```

Tell WEB•FM what action to perform by naming the submit pushbutton "Add".

Example:

```
<INPUT TYPE="submit" NAME="Add"
VALUE="submit">
```

It's always a good idea to target a specific database layout for creating a new record. Performance will be improved because WEB•FM will only have to interface with a small subset of database fields (those on the target layout) and not all fields in the entire database. The layout variable you place on your fill-out FORM will look something like this:

```
<INPUT TYPE="hidden" NAME="layout"
VALUE="layout name">
```

It's important to include on your form the HTML variable for specifying a Template file containing a response page. In the case below, WEB•FM will return a Template file for a detail confirmation page with record values dynamically substituted.

Example:

```
<INPUT TYPE="hidden" NAME="html"
VALUE="detail.html">
```

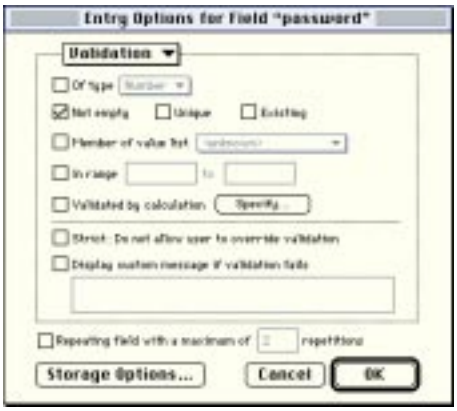
The complete submission form might look like this:

```
<FORM METHOD="POST" ACTION="EdExpert.fm ">

<INPUT TYPE="hidden" NAME="layout"
VALUE="data entry">

<INPUT TYPE="hidden" NAME="html"
VALUE="detail.html">

Name: <INPUT NAME="first Name"
VALUE=""><P>

Company: <INPUT NAME="Company"
VALUE=""><P>

<INPUT TYPE="submit" NAME="add"
VALUE="submit">

</FORM>
```

Having created the two most basic (and most often used) forms, let's take a look at some ways WEB•FM allows you to include greater functionality.

Support for Data Validation

Performing data validation on submitted field values (before a new record is created) is limited to checking whether the submitted value for a field is empty or not. To enable data validation, go to “Define Fields...” in FileMaker Pro and open the Data Validation dialog for the desired field and check the “Not empty” option. You MUST also enable the “auto-enter data” option for this field and have it automatically enter a space, otherwise an error will likely occur.



Should an INPUT element be submitted with no value when data validation for this field is enabled in the database, WEB•FM will not create the record and instead return a basic validation error message.

Support for Checkboxes

A form may contain multiple INPUT elements with the same name. Multiple checkboxes, for example, often have the same name but different values. When creating a new database record, WEB•FM will enter the values of fields with the same name into the database field with return delimiters so that FileMaker Pro can display the information as checkboxes on the database layout. The exception is with repeating and portal fields.

Support for Repeating Fields

If a database field is formatted as a Repeating field, WEB•FM will enter the values of fields with the same name into the repeating cells. A field defined as repeating in the database, but formatted on the target layout as a normal field will receive all values only in the first repeating cell with return delimiters. Make sure there are enough repetitions for the submitted data or an error will occur.

Support for Relational/Portal Fields

WEB•FM will add information to relational and portal fields, provided a target layout is specified using the Layout variable, and the name of the INPUT element on the submitted form is named properly. The name of the field should equal the name of the database relationship followed by two colons "::" and ending with the actual database field name in the related database.

Example:

```
<INPUT NAME="relationship::field name"
VALUE="">
```

If multiple values for a portal field are received, WEB•FM will create a new portal row for each value.

Support for Incoming HTTP Request [Tokens]

[username]	Username for current authentication scheme
[password]	Password for current authentication scheme
[browser]	Client Browser
[domain]	Domain name or IP address of client
[referer]	Referring document
[cookie]	Use with CookieCutter Processor to insert client "cookie" values in HTTP request header into the request form arguments.

Example:

```
<INPUT TYPE="hidden" NAME="browser"
VALUE="[browser]">

<INPUT TYPE="hidden" NAME="domain"
VALUE="[domain]">

<INPUT TYPE="hidden" NAME="customer"
VALUE="[cookie]">
```

Editing A Database Record

WEB•FM 4.0 significantly enhances features for record-level security. A revised FindUser command supports most Find command features, but with record-level user authentication requirements. This means users can flexibly Find and modify their own, and only their own, records in databases where the permission to Browse records is disabled. Moreover, a client Cookie value may also be used in place or absence of a valid password when finding or updating database records.

Use the FindUser command to return an update form with existing record information. By using FindUser, this update form is not returned unless the user has a valid username and password, or browser "cookie" value.

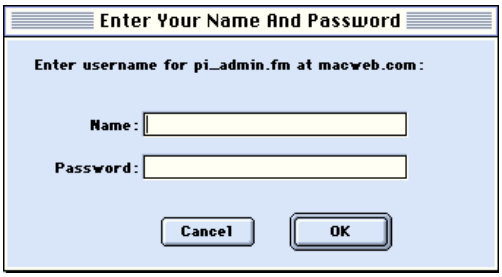
Example:

```
<A href="EdExpert.fm$FindUser?
layout=data+entry&html=update.html">
```

If you expect more than one record to be found with FindUser, you should return a "hit list" of found records from the database first, and then select the desired record from this list using a record number identifier.

Example:

```
<A href="EdExpert.fm$FindUser?RecNum=2
&layout=data+entry&html=update.html">
```



Use the Update command on a form if you wish to modify an existing database record with submitted form values. The Update command is equivalent to the Add command, only a new record is not created. Instead, WEB•FM performs a find for a single record, then verifies authorization to update this record. If authentication checks pass, submitted form values will replace existing cell values in the found record.

W A R N I N G ***Submitting INPUT elements with values that are empty will DELETE existing database cell values.***

To Update an existing database record WEB•FM must first find the correct record to update. An update form should contain the following INPUT variables. If no Field and Value variables are specified, WEB•FM still attempts to find the correct record using the username and password, or browser cookie value. In this case, though, [RecID] is a token which gets replaced with the record identifier when the update form is initially retrieved. You may use RecID or the name of another serial number field for the unique record identifier.

```
<FORM METHOD="POST"
ACTION="Database+Name.fm ">

<INPUT TYPE="hidden" NAME="field"
VALUE="RecID">

<INPUT TYPE="hidden" NAME="value"
VALUE="[RecID]">

<INPUT TYPE="hidden" NAME="layout"
VALUE="layout name">

<INPUT TYPE="hidden" NAME="html"
VALUE="detail.html">

<INPUT TYPE="submit" NAME="update"
VALUE="submit">

<FORM>
```

Unless you have the administrator password, it is not possible to edit an existing record without first creating new "username" and "password" (or "cookie") fields. Authorization to update is given if the request passes one of the following checks.

1. Browser username/password exactly matches Admin username/password.
2. Browser username/password exactly matches record username/password.
3. INPUT username/password exactly matches record username/password.
4. Browser cookie value exactly matches record cookie.

Be very careful when updating portal fields. WEB•FM will replace existing portal field values with new values in the order sent by the web browser. There is real potential to edit an incorrect portal row, so test carefully with different browsers before deploying a solution that requires editing existing portal rows.

Page Generation 7

Template File Support

WEB•FM 4.0 introduces support for easy-to-use Template files. Template files are Internet standard HTML documents void of confusing, incompatible, and proprietary markup [/tags]. For this reason a webmaster need not learn yet another markup language, and Template documents are fully compatible with all existing web page editors and client browsers.

A highly intelligent parsing algorithm in WEB•FM automatically selects appropriate pop-up/pull-down menu items and automatically checks appropriate radio button or checkbox fields in a Template document using database record information. A single context-sensitive [field name] token may be used throughout a Template document to dynamically insert cell data where desired, auto-populate pop-up/pull-down menus, and auto-replicate radio button and checkbox fields from database value lists, all with no tags required!

Example Template:

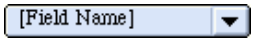
Template files must reside in the web server directory specified by the current request URL. Template file names must end with a ".html" suffix. When referred to with the HTML variable, the ".html" suffix is a flag for WEB•FM to look in the current directory for the Template file and perform database field and value list substitutions. Selection of existing pop-up menus, checkboxes, and radio buttons in a Template document are automatic.

Example URL:

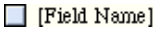
I M P O R T A N T ***Template files may only be used when returning information from a single database record.***

[Tokens] are considered place holders for substituting dynamic data. Square [brackets] surrounding a [field name], [search field], [variable], or other [value] help WEB•FM perform dynamic substitutions (i.e. server-side includes).

[Field Name] tokens are context-sensitive. That is, where [field name] tokens are used in a Template file influences substitution behaviour. By itself, a [field name] token is replaced with raw cell data from the found record. When present in a pop-up/pull-down menu, [field name] will auto-populate menu items using values from a database value list assigned to that field on the target layout.



When flush against an HTML checkbox or radio button, WEB•FM replicates the INPUT element with "
" delimiters for each item in the field's value list. The actual NAME attribute of pop-up menus, checkboxes, and radio buttons is irrelevant as far as substitutions are concerned.



Syntax Examples:

Dynamic database [field name] substitutions:

```
[field name]
```

Dynamic population of pop-up menus and scroll lists:

```
<OPTION>[field name]
```

Dynamic replication of checkboxes and radio buttons:

```
<INPUT TYPE=”checkbox” NAME=”field name”
VALUE=””>[field name]
```

I M P O R T A N T ***Dynamic population of pop-up menus and dynamic replication of checkboxes or radio buttons requires the field on the target layout be formatted with a database value list.***

Portal Fields

WEB•FM supports a custom <portal> tag for dynamic replication of enclosing HTML for each portal row of a relational field on a layout. Template files with [portal field] tokens require this tag so WEB•FM can replicate portal rows using the desired HTML data. Both [field name] and [portal field] tokens are legal inside the <portal> tag.

Remember from a previous chapter that relational fields must be referred to by a special name that begins with the name of the database relationship, followed by two colons “::” and the actual field name.

Syntax Example:

```
<portal><LI>[relationship::field
name]<P></portal>
```

Error Template Files

The Error variable supports Template files in a similar fashion as the HTML variable. However, rather than [field name] tokens we use [search field] tokens instead. WEB•FM will select appropriate pop-up/pull-down menu items and automatically check appropriate radio button or checkbox fields in a Template file using submitted find field criteria. The [search field] token may be used throughout a Template document to dynamically insert cell data where desired. WEB•FM uses submitted FORM arguments for field substitutions when an unsuccessful Find returns no found records.

Understanding [Token] Support

[Tokens] are considered place holders for substituting dynamic data. Square [brackets] surrounding a [field name], [search field], [variable], or other [value] help WEB•FM perform dynamic substitutions (i.e. server-side includes) in the Template file for dynamic page generation.

Outgoing server-side include substitution [tokens]

[RecID]	The unique ID for a record maintained internally by the database. Limited to one [RecID] token per record.
[found]	An integer value for the number of records found.
[values]	A text string of query values with comma delimiters.
[fields]	A text string of query fields with comma delimiters.
[date]	The current date.
[start]	The first record number currently displayed in the found set.

[end]	The last record number currently displayed in the found set.
[max]	The integer value for total number of records displayed.
[range]	The first and last record number for the next set of found records.
[page]	The current page number.
[pages]	The total number of pages (influenced by [max]).
[database]	The current database name.
[layout]	The current database layout.
[sort]	Value of current Sort variable.
[sortorder]	Value of current SortOrder variable.
[script]	Value of current DoScript variable.
[html]	Value of current HTML variable.
[field name]	Database cell value.
[portal field]	Database relational field value.
[search field]	Query field value.

Example:

```
Page [page] Results: [start] - [end] of
[found] found by [sort]
```

Understanding <Tag> Support

<next></next>	Custom tag for inserting a link to the next subset of found records.
<prev></prev>	Custom tag for inserting a link to the previous subset of found records.
<portal></portal>	Custom tag in Template files for replication of portal rows.

Example:

```
<next>Next [max] Results ([range])</next>
```

Calculation Formulas, Functions and Uses

WEB•FM 4.0 relies partly on Template files and partly on database calculation fields to dynamically build and deliver HTML documents for publication on the web. Template files are easier to use and design, but HTML calculation fields can be far more powerful. In the case of calculation fields, WEB•FM's job is to grab HTML from these fields and serve them to the web browser. By using a calculation field you have complete control over the HTML formatting of your documents, including direct access to over 100 powerful calculation functions supported by FileMaker Pro. Keep in mind that only the "html" calculation field is required for the purpose of returning 2 or more found records. Refer to Chapter 3: Getting Started for a short tutorial on creating calculation fields that build HTML documents. Using calculation fields will also offer slightly better performance.

USEFUL FUNCTIONS

If (test, result one, result two)

IsEmpty (field)

GetRepetition (repeating field, number)

PatternCount (text, search string)

Substitute (text, search string, replace string)

NumToText (number)

DateToText (date)

Status (CurrentDate)

Status (CurrentTime)

Status (CurrentRecordNumber)

Returning a Random Banner

Here's a useful technique for cycling through a set of custom headers with each and every WEB•FM database transaction. This technique might be used, for example, to show a different advertiser on each and every database hit, or change the background color of an HTML page randomly.

1. ***Create a field in your database named "header" of type "calculation".***
2. ***Set the Storage Options for this field to NOT store results and to "evaluate only when needed".***
3. ***Use the Choose and Random functions as shown below to dynamically return a different result each time the calculation evaluates.***

In this example, we're returning a random image from a selection of 5 different choices. Any database hit that uses the Header variable, the formula will evaluate each time to generate a different result. The Find and FindAll commands, remember, have a default to return the "header" field automatically if it exists.

```
"<IMG SRC="" &
Choose(NumToText(Int(Random*10)),
"zero.gif","first.gif", "second.gif",
"third.gif", "fourth.gif",
"zero.gif","first.gif", "second.gif",
"third.gif", "fourth.gif") & "">"
```

Expiring HTML Documents

This formula for expiring HTML is useful for protecting private information from hackers, but momentarily making the information available to the person who creates a record.

If(Status (CurrentDate) = CreateDate and Status(CurrentTime) - CreateTime ≤ Time(0, 0, 5), "html with private password", "unauthorized")

Processors 8

WEB•FM 4.0 introduces a new plug-in architecture for database level add-on Filter and Server-Side Include (SSI) Processors such as Maxum Development's NetCloak 2.5. Processors are separate web server plug-ins supporting the PIXO ("Plug-In Cross-Over") API standard for web server plug-ins to communicate and cooperate.

The value of Processors are that they offer extensibility to the capabilities of WEB•FM. Custom security or logging processors are good examples where a processor might prove useful. Off the shelf processors even exist like NetCloak 2.5. Keep these points in mind:

- Processors must be PIXO compatible.
- Filters process incoming requests.
- SSI process outgoing reply pages.
- Processors apply at database level.
- Processors may be chained.
- Chain determined by selection order.

Installation

Processors are stand-alone plug-ins and need to be installed in the web server's "Plug-ins" folder. Plug-in services are named according to the ACTION they register at web server startup. These ACTION names must to be defined using the Plug-in services pop-up menu in the Admin database (shown below) before they can be called by WEB•FM.

WEB•FM includes Processor services pre-defined for "cookie" support and international character translations, among others. To enable a Processor, drag the plug-in into the web server's "Plug-ins" folder, restart the server, and update the Admin database with the plug-in service enabled for the desired database.

ENCODER PROCESSOR

ASCII is one encoding system for the standard characters in the range of zero to 127. Characters from 128 to 255 are defined differently for different operating systems, different languages, and different countries. Because of these differences, computers may encode values differently so they will not translate correctly without help.

The Encoder Processor plug-in offers the capability to perform character encodings on out-going HTML documents using a customizable character translation table. The Encoder plug-in can be called upon by WEB•FM to perform character encoding for improved international browser support.

A common character set for many types of software is the ISO Latin 1 character set as defined in the ISO 8859-1 (1987) standard. Included with the Encoder plug-in is an “Encoding Table” for translating from the MacRoman character set to the ISO Latin 1 character set. But because this table will not work for all countries, you can and should modify this table for your specific translation needs. Translation tables go into the “FM Prefs” folder in the server’s “Plug-ins” folder. The Encoder plug-in expects the file to have a name of “Encoding Table”.

A FileMaker Pro database is included with pre-defined entries for character encodings. Use this database to find and omit character encodings as you need. Use the “Export...” menu to export a new “Encoding Table”.

T I P ***Refer to the chapter on Advanced Features for information on including custom HTTP response headers that include fields for specifying the “Content-Type” language for encodings.***

DECODER PROCESSOR

The character mapping problems that exist between different operating systems and languages with regard to outgoing documents also exists for incoming requests. An accented character submitted with a Find or Add web form may not end up as the same character when viewed in the database. For this reason the Decoder plug-in can be installed with optionally a custom “Decoding Table” and called by WEB•FM for character translation services.

REPLACE PROCESSOR

The Replace plug-in is an additional Processor for miscellaneous search & replace services. Install and customize the “Replace Table” as needed.

FILTER PROCESSOR

The Filter Processor is pre-configured with a number of search & replace strings for incoming requests. Enable this Processor to use abbreviated, succinct URL’s or to perform data manipulation tasks. Example substitutions might include:

“sort=down”	“sortorder=descending”
“sort=up”	“sortorder=ascending”
“&script=”	“&doscript=”
“op=”	“operator=”
“/00”	“/2000”
“filter=today”	“date=//&sort=date&sortorder=descending&max=10”
“filter=all”	“sort=date&sortorder=descending&max=10&findall=”
“filter=add”	“client=[domain]&browser=[browser]&referer=[referer]”

COOKIECUTTER PROCESSOR

Included as a component of Got Cookies?, “CookieCutter” has a singular purpose to filter incoming database request arguments, replacing [cookie] tokens, if any, with the appropriate “cookie” value in the HTTP request header. The “cookie” MUST have a name equal to the database field you intend to use it with. This is because CookieCutter looks at the name portion of the incoming name/value pairs to determine which HTTP cookie value needs to

replace the [cookie] token. In this example, the two [cookie] tokens are replaced by the corresponding cookies “user” and “referal” (if any) in the HTTP request header.

Example:

```
“user=[cookie]&referal=[cookie]”
```

Refer to Got Cookies? for information on how to set client-side cookie values.

Security 9

Naming Your Database

Because it is possible for a web server to return an entire FileMaker Pro database directly from disk, the common rule of thumb is to place the database outside the web server's directory tree. On the other hand, it's generally more convenient to keep the database, Template files, images etc. in the same solution folder. With WEB•FM, this is not a security problem provided the database name ends with a ".fm" suffix. Whenever the web server sees ".fm" as the suffix for a requested file, it automatically passes the request to WEB•FM for processing rather than serving the database from disk. The ".fm" at the end of the file is known as a suffix. It is the default suffix mapped to WEB•FM. You can change this suffix within you web server administration package. If your database name does not contain a suffix, place the database outside of the server folder. If it does, you are perfectly safe keeping the database in the server folder along with any external HTML documents you might be using.

Securing the PI_ADMIN.FM database

The database “pi_admin.fm” is your security control panel for ALL of the databases you publish online. It’s the interface you use to administer the defaults and security settings used by WEB•FM. Proper care should be taken to prevent access to this database and it’s contents. The Admin password for a specific database provides FULL access to that database regardless of the specified permissions and field security settings. The Admin password for the “DEFAULT” database entry provides FULL access to any database which does not have an entry. If the Admin password for a database is blank, this enables FULL access privileges to that database, including updating and deleting database records. This may actually prove useful in an Intranet setting or other situations where security may not be an issue but full access to the database is.

Here are several suggestions for securing access to the “pi_admin.fm” database.

1. Define a web server realm with “pi_admin” as the match string if one is not already defined. If you wish, you may add a web server password that allows access to this realm. Most likely the realm password you choose is the same password you entered for the “pi_admin.fm” database entry during installation. You can of course make the passwords different so not even the administrator has web access to the Admin database.
2. It’s not required that the Admin database remain open or even exist on your web server, although it can prove convenient.
3. The Admin database supports remote LAN administration of the settings file used by WEB•FM. This means you can remove the Admin database entirely from the web server and administer WEB•FM remotely if you wish from another computer using Program Linking.

4. If you enable Program Linking on your web server, be sure to disable Guest access so someone else on the LAN can't somehow update the settings remotely.
5. The Admin database itself may also be password protected on open. To enable password protection, open the Admin database and from the File menu select "Change Password…". In the dialog that appears, enter "WEB•FM" as the old password, then enter your new password.

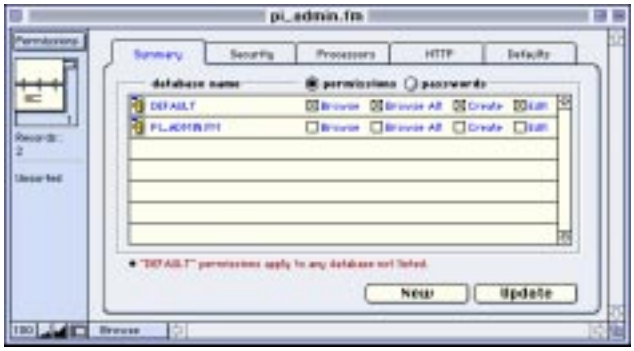
Available Security Options

Database-level, Realm-based Security

Because each and every database transaction using WEB•FM requires the name of the target database be specified in the URL, you can setup a secure web server realm with your database name as a match string. This will implement password protection at the web server level for all web activity on the specific database.

Task-level Security

Use the Admin database to optionally disable or enable tasks as appropriate for the needs of your specific database. A valid Admin password may be used to override these restrictions.



Available commands fall under the following task permissions:

Browse Records	Find, Retrieve, Random
Browse All	FindAll
Create Records	Add
Edit Records	Update, Delete, FindUser

Field-level Security

Use the Admin database to optionally disable access to specific database fields. In the “Finding” field enter any field name you wish to restrict from someone performing a Find. In the “Returning” field enter any field name you wish to restrict from someone returning data. A valid Admin password may be used to override these restrictions.

Record-level Security

A database where users need to update and delete existing records requires that the database contain “username” and “password” fields or a “cookie” field. If the database does not contain these fields the only way to modify existing records is by supplying the Admin password for that database.

Configuring a Custom Suffix Mapping

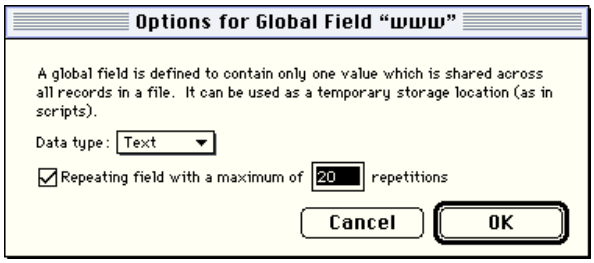
On startup, WEB•FM automatically defines a web server suffix mapping entry for ".fm". Any HTTP request for a file ending in ".fm" will automatically be mapping to the WEB•FM action for processing, which is also automatically defined on startup.

It may be useful to manually enter and use a web server suffix entry other than ".fm". Perhaps the names of your databases have a different suffix (".fp3" for example), or you might prefer a suffix which better markets your online presence (".orders" for example). To do this, simply define a new web server suffix entry mapped to the WEB•FM action. If the name of the target database does not actually have the suffix used, WEB•FM will automatically strip the suffix from the file request.

Using HTTP Request Parameters

WEB•FM provides a way to temporarily capture to the target database many of the client HTTP request parameters received by the web server. These parameters may prove useful in some way for scripts or calculation fields which you may have in the database. They are also useful in implementing intelligent server-side include features without having to rely on another web server add-on. There can be a small performance hit by using this feature on older model servers, so if you don't use this feature do not include the "WWW" field in your database.

Create a new Global field in your database named "WWW" and give it 20 repeating cells.



Upon every transaction WEB•FM will place 20 HTTP request parameters into the repeating cells. Using a script or calculation and the GetRepetition() function you can access a specific cell for the desired parameter and take action based upon that parameter. The first cell can normally be referred to by field name alone (WWW) without using the GetRepetition() function since it's the first repetition.

The order in which repeating cell values get data is as follows:

1.	FormData	Submitted form data, or search argument portion of URL
2.	Method	HTTP method (e.g. GET, POST)
3.	Domain	Domain name of client
4.	Username	Username for current authentication scheme
5.	Password	Password for current authentication scheme
6.	Email	“From” header field (email address when supplied)
7.	ServerName	Domain name of server
8.	ServerPort	Port number of server
9.	URLPath	URLPath
10.	Command	Database command (or Path argument portion of URL)
11.	Referer	Refering document
12.	Browser	
13.	Realm	
14.	SearchArg	
15.	ClientIP	
16.	FullRequest	
17.		
18.	TotalConnections	
19.	CurrentUserLevel	
20.	HighestUserLevel	

As an example for how the “www” field might be used, you can tailor your HTML documents to the specific browser of the client. By retrieving the HTTP data that tells you what browser the client is using, you can send a Retrieve command to your FileMaker Pro database that will deliver the appropriate HTML for each specific browser.

Returning a Custom HTTP Response Header

If the reply document (Template file or HTML field) begins with a raw HTTP header, WEB•FM does NOT prepend an HTTP response header to the out-going HTML. You are responsible for building a valid HTTP response header properly formatted with the appropriate number of returns and line feeds.

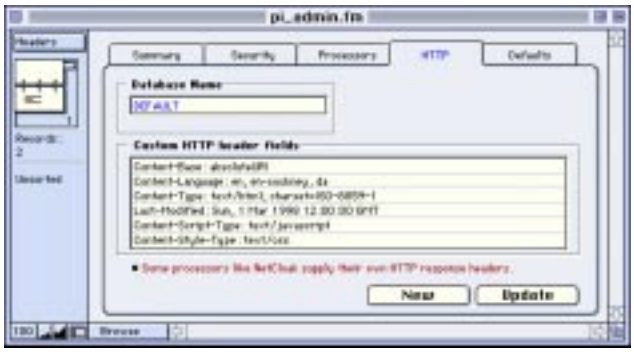
Advanced uses might include sending a URL redirect to the client browser for sending an email message using another plug-in, or setting a browser "cookie" value for an online shopping cart complete with expiration dates.

Example:

```
HTTP/1.0 302 Found

Location:
cookie.fm$Add?cookie=[cookie]&html=detail.html

Set-Cookie: CUSTOMER=WILE_E_COYOTE; path=/
; expires=Tuesday, 31-Dec-96 23:12:40 GMT
```

Custom HTTP Header Fields

The Admin database includes a tab (shown below) for specifying custom HTTP response header fields. Entries are appended to the HTTP header WEB•FM sends to the browser in reply to a request. Custom HTTP fields are applied at the database level for every request. This has the advantage of not requiring some <META> tags in every HTML document.



Use a Content-Type field for better international character support in conjunction with the Encoder and Decoder Processors.

Use a Last-Modified field to force the client browser to locally cache database transactions. For example, you might create a small, simple database with the sole purpose of serving Find forms with dynamic population of pop-up menus from a changing database value list. In this way, an index page of some kind could be referred to throughout a web site and not require another database hit after the first request.

APPENDIX

Optimizing Performance

Web Server Tips

- A properly configured web server has only the bare minimum of System Extensions and Control Panels installed or enabled.
- Avoid using File Sharing on your web server machine unless you need to.
- Configure the Memory Control Panel with perhaps a largish 2024K disk cache size.
- Disable your web server's option for reverse DNS lookup.
- The web server should DEFINITELY run as the frontmost application.
- Avoid running unnecessary additional programs on your web server.
- If you run additional programs on your web server, do some performance comparisons with these other applications both running and not running. Some ftp software, for example, can *severely* affect web server performance. Most web servers have a transaction duration parameter (tick count) which can be enabled for comparison purposes.
- Run the most recent versions of Open Transport, your web server software, and FileMaker Pro.

FileMaker Pro Tips

- Keep FileMaker Pro a hidden background application. If a database layout is visible during a database request, than screen redraw of a database window can *severely* affect database performance. Here are three options for hiding database windows:
  1. Select "Hide Others" from the application menu.
  2. Use WindowShade to double-click and display only a title bar for a database window, thus hiding the current database layout.
  3. Make the default database layout sparse with graphics and database fields.

- Allocate a sizeable amount of free RAM to the FileMaker Pro application. 8MB is good, 16MB or more may be better.
- Consider the risks and benefits of using a RAM disk for the actual database file, since FileMaker Pro tasks are disk intensive.
- Store and open your database from the hard drive or RAM disk of your local machine. In other words, avoid opening the database as a client of another machine hosting the database as this will add noticeable network overhead.
- Avoid enabling multi-user mode for your databases unless you need to.
- When adding or updating database records, specify a target layout for the transaction. This may require that you modify your database with a new layout and fewer fields. The fewer the fields on a target layout there are the fewer fields that must be dealt with.
- Database calculation fields which evaluate HTML should normally have their Storage Options set to "Do not store calculation results — calculate only when needed". This is especially important if your web solution creates new database records, since you do not want FileMaker Pro to evaluate the formula again and again on each field update. One exception might be the "html" field and any other fields which are generally returned as a hit list of found records. Certainly never index an HTML calculation field unless you intend to use the index for a value list.
- ScriptMaker scripts invoked via a web browser should be visible in the Script menu.
- Periodically use FileMaker Pro's command to "Save a compressed copy" of your database.

Available Commands

Find Equivalent to selecting Find from the Mode menu in FileMaker Pro and performing a Find for entered query values. Returns a list of found records to the web browser with a header and footer, or an error page if no records are found.

FindAll Equivalent to selecting Find All from the Mode menu in FileMaker Pro. Returns a list of all records in the database, or optionally a value list of only all the unique records in the database.

FindUser Supports most Find command features, but with record-level user authentication requirements using the username and password values (or HTTP cookie value) from the client browser.

Retrieve Equivalent to selecting Find from the Mode menu in FileMaker Pro and performing a Find for entered query values. Returns a single detail record with no header or footer unless one is explicitly specified.

Random Returns a random detail record from the database.

Add Equivalent to selecting New Record from the Mode menu in FileMaker Pro and entering new cell values.

Update Similar to performing a Find in FileMaker Pro based on the System setting for current User Name. Modifies found record with submitted FORM data if username and password (or cookie value) pass authentication checks.

Delete Equivalent to selecting Delete Record from the Mode menu in FileMaker Pro. Similar to performing a Find in FileMaker Pro based on the System setting for current User Name. Deletes found record with submitted FORM data.

Available Variables

Header Specifies a database field to return as the header above a found record or list of found records. May be used with any command. If omitted from a FORM and the command is Find or FindAll, then the default is to return the "header" field from the database.

Footer Specifies a database field to return as the footer below a found record or list of found records. May be used with any command. If omitted from a FORM and the command is Find or FindAll, then the default is to return the "footer" field from the database.

Error Specifies a Template file or database field to return when there is a problem performing a database transaction. May be used with any command. If omitted from a FORM and the command is Find or FindAll, then the default is to return the database "error" field when no records are found.

HTML Specifies a Template file or database field to return as the main body of an HTML document. Required for every command, but may be omitted from a FORM or hypertext link if the default to return the "html" database field is desired.

Operator Specifies how a database Find should "operate". If omitted from a FORM, the default "begins with" is assumed. Operator is not field specific but applies to the entire FORM. Both comparison and logical operator values are available.

begins with	less than or equal
contains	greater than
ends with	greater than or equal
equals	request
not equal to	or
less than	

Field Specifies a database field name to search on. This variable also requires the Value variable which tells WEB•FM what to search for. Field/Value may be used with most commands to perform special needs. Normally used with the Find command so the desired search field can be user selectable. Mostly used with the Update and Delete commands to specify which record should be edited.

Value Specifies a query value. Required when using the Field variable.

Sort Specifies the database Sort field for the found records. Two-level nested sorts are recognized by specifying two Sort variables.

SortOrder Specifies the direction for sorting found records. Requires also using the Sort variable. Recognizes "ascending" and "descending" as valid options. If omitted from a FORM, than "ascending" is the assumed default.

Max Specifies a maximum number of records to return from a found set. Value of Max must be an integer. If omitted, then all found records will be returned.

Layout Specifies the name of a database layout to use when performing a database transaction. Recommended on a FORM for use with the Add and Update commands or when returning a Template file. Layout is required when dealing with ANY relational fields in any manner.

DoScript Specifies the name of a ScriptMaker script to run. The DoScript variable will invoke a ScriptMaker script in a database just prior to returning HTML. Scripts should be visible in the Script menu, otherwise WEB•FM will momentarily toggle FileMaker Pro to the foreground.