For Windows, Mac OS X Server, and Red Hat Linux

FileMaker® Pro 5.5 Unlimited

Administrator's Guide

©1999, 2001 FileMaker, Inc. All Rights Reserved.

FileMaker, Inc.
5201 Patrick Henry Drive
Santa Clara, California 95054
www.filemaker.com

FileMaker is a trademark of FileMaker, Inc., registered in the U.S. and other countries, and the file folder logo is a trademark of FileMaker, Inc. Red Hat is a registered trademark of Red Hat, Inc., in the U.S. and other countries. All other trademarks are the property of their respective owners. Portions of some screen shots are reprinted by permission from Microsoft Corporation. Portions of some screen shots are copyright 1996-2000 Netscape Communications Corp. All rights reserved. These screen shots may not be reprinted or copied without the express written permission of Netscape.

This software is based in part on the work of the Independent JPEG group.

Portions of this software are copyrighted by MERANT. All rights reserved.

All persons and companies listed in the examples are purely fictitious and any resemblance to existing persons and companies is purely coincidental.

Mention of third party companies and products is for informational purposes only and does not constitute an endorsement. FileMaker assumes no responsibility with regard to the selection, performance, or use of these products. All understandings, agreements or warranties, if any,take place directly between the vendor and prospective users.

This page intentionally left blank.

Contents

Chapter 4
Custom web publishing using CDML

Chapter 5
Using FileMaker Pro XML to deliver your data

Chapter 6
Using Java and JDBC to deliver your data

Appendix A
Valid names used in CGI requests for FileMaker XML data

Preface
Introducing FileMaker Pro 5.5 Unlimited

FileMaker® Pro 5.5 Unlimited is a companion product to FileMaker Pro 5.5 that allows unlimited web user access to your FileMaker databases via the Web Companion plug-in.

In addition, the FileMaker Pro 5.5 Unlimited product includes the following features:

- The FileMaker Web Server Connector that allows you to combine the web serving features of the FileMaker Pro Web Companion with the features of these web servers: Microsoft Internet Information Server 4.0 and Microsoft Internet Information Server 5.0, Apache on Mac OS X Server, and Apache on Red Hat® Linux 7.1.
- FileMaker Pro custom web publishing tools and example files to help you publish your database on custom web pages using CDML, XML, or a Java applet.

Note For deploying your custom database solutions over a network, the FileMaker, Inc. product line also includes FileMaker Server. FileMaker Server provides multi-protocol support for TCP/IP, IPX/SPX, and AppleTalk networks for serving up to 125 hosted files simultaneously to FileMaker Pro guests on Windows 2000, Windows NT, Mac OS 8.6 to 9.1, Mac OS X, and Red Hat Linux platforms.

About this guide

This *Administrator's Guide* provides instructions and examples for using FileMaker Pro 5.5 Unlimited.

- This preface describes the contents of the FileMaker Pro 5.5 Unlimited CDs.
- Chapter 1 describes how to install the Web Server Connector on the Microsoft Internet Information Server, Apache on Mac OS X Server, and Apache on Red Hat Linux web servers.
- Chapter 2 provides information on configuring the administration account and administering the Web Server Connector from your web browser.
- Chapter 3 gives information on setting up and using the Web Companion for creating custom web pages.
- Chapters 4 and 5 and appendixes A and B describe the use of the FileMaker Pro Web Companion for *custom web publishing* using XML or CDML.
- Chapter 6 describes how to use the FileMaker JDBC Driver to create FileMaker Pro database-aware Java applications and applets. It also includes a section on using the proprietary FileMaker Java Class Library.
- Appendix C describes how to enable the Web Server Connector in Mac OS X Server.

For information on installing and configuring Internet Information Server on Windows 2000 Server, Apache on Mac OS X Server, or Apache on Red Hat Linux, see the documentation provided with your web server software.

For information on authoring and using XML, creating Cascading Style Sheets (CSS) or Extensible Stylesheet Language (XSL) stylesheets, and authoring web pages in Dynamic HTML (including scripting such as JavaScript), see the documentation that came with your authoring tool.

Contents of the FileMaker Pro 5.5 Unlimited CDs

FileMaker Pro 5.5 Unlimited consists of two CDs: one for the FileMaker Pro software and related files, and one for custom web publishing and the Web Server Connector.

Contents of the FileMaker Pro CD

The *FileMaker Pro* CD provides the installer for the FileMaker Pro 5.5 Unlimited software. For installation instructions and for information on the minimum hardware and software requirements for using FileMaker Pro 5.5, see the printed *FileMaker Pro 5.5 Getting Started Guide* or the *FMP 5.5 Getting Started.pdf* document on this CD.

The installation code that comes with FileMaker Pro 5.5 Unlimited is the key to unlimited web user access to the Web Companion.

FileMaker on the Web links

Double-click FileMaker on the Web to open the Go_FileMaker_Unlimited.html page in your browser. Then, click the link to go to the web site. There you will find product information and support, as well as helpful links to other resources.

Contents of the FMWSC and Tools CD

The following tables describe the contents of the *FMWSC and Tools* CD.

CDML folder

Contents of the CDML folder	Description
CDML Examples	Folder containing three folders of example files: guest_book folder, employee_database folder, and shopping _cart folder For information about these examples, see chapter 4, "Custom web publishing using CDML."
CDML Templates	Folder containing nine CDML format files for different types of interaction with a database
Web Tools	Folder containing the CDML Reference.fp5 and CDML Tool.fp5 databases and the CDML Templates folder

Custom Workgroup Portal folder

Contents of the Custom Workgroup Portal folder	Description
Asset Management.fp5	Sample FileMaker Pro database
Personnel Records.fp5	Sample FileMaker Pro database
customwebportal.mth	Custom Workgroup Protal example file
collage3.jpg	Graphic file
logo.gif	Graphic file

Electronic Documentation folder

The Electronic Documentation folder contains a PDF (Portable Document Format) version of the *FileMaker Pro 5.5 Unlimited Administrator's Guide*, which you can print from Adobe Acrobat Reader.

FileMaker JDBC Driver folder

Contents of the FileMaker JDBC Driver folder	Description
Java Class Library	Folder containing Java Documentation and Examples
JDBC Documentation	Folder containing HTML based documentation for JDBC
JDBC Examples	Folder containing examples for FileMaker Explorer, JBuilder 3.0 Professional, and Visual Cafe 4.0 Expert Edition

Contents of the FileMaker JDBC Driver folder	Description
fmpjdbc12.jar	The FileMaker JDBC driver
Read Me.doc	JDBC read me file

Unsupported folder

Contents of the Unsupported folder	Description
FMWSC.jar	The FileMaker Web Server Connector servlet
FMWSCResources.jar	Strings and supporting HTML pages for the Web Server Connector servlet
Readme.txt	Text file containing instructions for recompiling the .jar file for newer versions of Apache

XML folder

Contents of the XML folder	Description
Documentation	Folder containing the XML Document Type Definitions (DTDs) for the FileMaker XML grammars: fmpdsoresult_dtd.htm fmpxmllayout_dtd.htm fmpxmlresult_dtd.htm
Inventory Example	Example of the inventory.fp5 database published on the Web using XML, the W3C DOM, and JavaScript Files include : inventory.fp5, default.htm, detail_view.htm, add_record.htm, find.htm, Filemaker.gif and FMP.js (JavaScript library for the example) For information on this example, see chapter 5, “Using FileMaker Pro XML to deliver your data.”
Simple Examples	Three examples of the People.fp5 database published on the Web using XML with CSS, XSL, and JavaScript Files include: default.htm, Filemaker.gif, FMP.js, people.fp5, people_form.css, people_form.xsl, and people_form.htm. For information about these examples, see chapter 5, “Using FileMaker Pro XML to deliver your data.”

Registration and customer support

Please complete and mail the registration card for your FileMaker product, or register online at www.filemaker.com/register. In FileMaker Pro, you can choose Help menu > FileMaker on the Web for a link to the FileMaker, Inc. support pages and the online registration form.

For information about technical support and customer service, see:

www.filemaker.com (North American customers)
www.filemaker.com/intl (customers outside of North America)

At the web site you will find the FileMaker, Inc. Service Directory, which provides the service options available to North American customers, links to FileMaker, Inc. international sites, answers to frequently asked questions, and access to extensive software libraries used by technical support staff.

If you do not have access to the Web, please refer to the Technical Support and Customer Service sheet included in the software box. North American customers can also call 1-800-965-9090 to learn about the service options available.

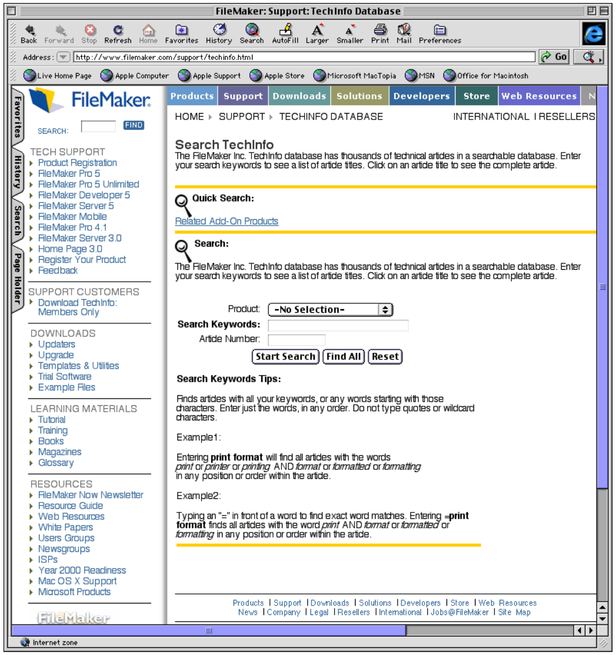
Where to go for more information

For more information on	See
New features in FileMaker Pro 5.5	FileMaker Pro Help or chapter 4 in the *FileMaker Pro 5.5 Getting Started Guide*
General information on Instant Web Publishing	Chapter 14 in the *FileMaker Pro 5.5 User’s Guide*
Web publishing enhancements	Chapter 4 in the *FileMaker Pro 5.5 Getting Started Guide*
Other resources	The FileMaker on the Web link on the *FileMaker Pro* CD

Search the TechInfo database

The TechInfo database is a great resource for technical information about FileMaker, Inc. products. This FileMaker Pro database serves as a front-line resource for the company’s support technicians as they field customer inquiries. It collects Q&As, tips, FAQs, bug reports, update notes, press releases, templates, and a host of other material valuable for the support professional.

The TechInfo database is available on the product support pages on the FileMaker, Inc. web site at www.filemaker.com.



FileMaker: Support: TechInfo Database

Address: http://www.filemaker.com/support/techinfo.html

HOME ▸ SUPPORT ▸ TECHINFO DATABASE

Search TechInfo

The FileMaker Inc. TechInfo database has thousands of technical articles in a searchable database. Enter your search keywords to see a list of article titles. Click on an article title to see the complete article.

Quick Search:

Related Add-On Products

Search:

The FileMaker Inc. TechInfo database has thousands of technical articles in a searchable database. Enter your search keywords to see a list of article titles. Click on an article title to see the complete article.

Product: -No Selection-

Search Keywords:

Article Number:

Search Keywords Tips:

Finds articles with all your keywords, or any words starting with those characters. Enter just the words, in any order. Do not type quotes or wildcard characters.

Example1:

Entering **print format** will find all articles with the words *print* or *printer* or *printing* AND *format* or *formatted* or *formatting* in any position or order within the article.

Example2:

Typing an "=" in front of a word to find exact word matches. Entering =**print format** finds all articles with the word *print* AND *format* or *formatted* or *formatting* in any position or order within the article.

Chapter 1
Installing the FileMaker Web Server Connector

The built-in Web Companion in FileMaker Pro 5.5 Unlimited acts as a CGI (Common Gateway Interface) application and as a web server by handling browser interactions with FileMaker Pro databases and serving by web pages and image files to the web browser.

You can extend this functionality by using the FileMaker *Web Server Connector* included with FileMaker Pro 5.5 Unlimited. Use the Web Server Connector with any of the following supported web servers:

- Microsoft Internet Information Server 5.0
- Apache 1.3.19 bundled with Mac OS X Server
- Apache 1.3.19 bundled with Red Hat Linux 7.1 i386

About the Web Server Connector

The FileMaker Web Server Connector is a Java servlet that relays FileMaker CGI requests for XML, CDML, and other dynamic content from your web server to one or more computers running FileMaker Pro Unlimited. This allows you to keep your database files separate from your static HTML pages, images, and other non-database files. It also prevents the web browser from making unnecessary requests to FileMaker Pro Unlimited.

More importantly, the Web Server Connector allows you to use a RAIC (Redundant Array of Inexpensive Computers) with multiple copies of FileMaker Pro Unlimited so that more web users can access your databases at a time.

With the Web Server Connector, you can:

- take advantage of other plug-ins and features of your web server, including Secure Socket Layer (SSL) protection and server-side includes.
- configure a RAIC to increase throughput and reliability.
- store static elements such as image files on the web server, and bypass the FileMaker Pro Web Companion for pages that do not need to interact with the database.

Configuring a RAIC to use with the Web Server Connector

The Web Server Connector uses one or more RAIC machines, with copies of FileMaker Pro Unlimited installed on each one. In addition, you can use FileMaker Server or FileMaker Pro 5.5 in your configuration as a backend database host using peer-to-peer networking for better load balancing and fail-over protection. (For information about these products, see www.filemaker.com.)

Note It is possible to run FileMaker Pro Unlimited on the same machine as the Web Server Connector and the web server software, but for performance reasons, it's not recommended.

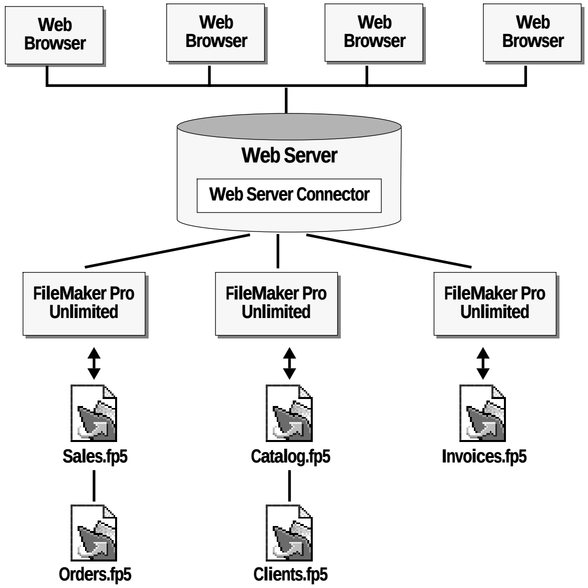
Here are some of the possible ways to configure your machines:

- The Web Server Connector on the web server machine and FileMaker Pro Unlimited hosting databases on another machine.

This is a strong configuration because you're able to realize all the server-derived benefits of the Web Server Connector using only two machines.

- The Web Server Connector on the web server machine and FileMaker Pro Unlimited on multiple machines, with each machine hosting different databases.

This configuration offers good throughput, as multiple hosts ease the load that would otherwise be given to a single host. It can provide a means for load balancing if the more active databases are divided between the machines.



Example of a RAIC with 3 machines (nodes in the RAIC) hosting multiple databases

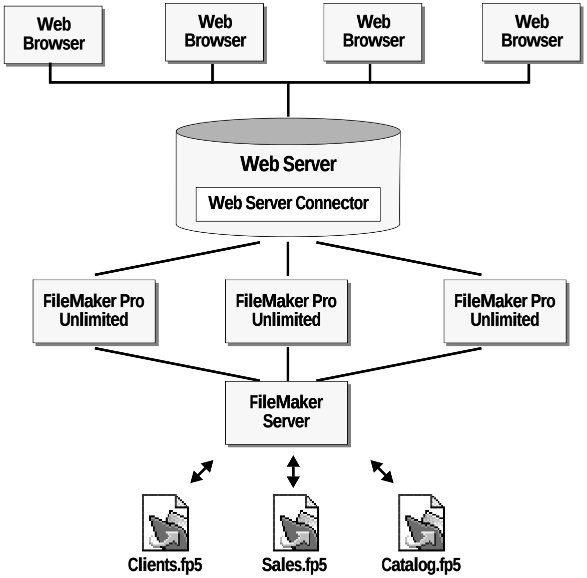
- The Web Server Connector on the web server machine and FileMaker Pro Unlimited on multiple machines, with each machine hosting a copy of the same read-only database(s).

This is useful for serving data such as a catalog. If a host crashes or is brought off-line, the Web Server Connector moves to the next host in the RAIC and web users are still able to access the database.

However, this configuration is not useful if you want to allow web users to write to and modify the database.

- The Web Server Connector on the web server machine, FileMaker Pro Unlimited on multiple machines, and FileMaker Server on a machine hosting the databases.

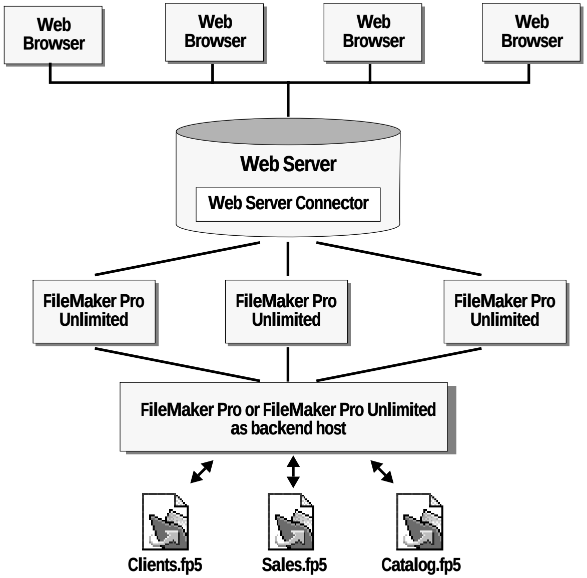
In this configuration, each copy of FileMaker Pro Unlimited accesses the databases as a guest of FileMaker Server. Web users can read and write to the databases, the Web Server Connector provides load balancing and fail-over protection, and FileMaker Server handles tasks more quickly by off loading functions such as sorting and summary field calculations to the guest machines. FileMaker Server also provides added data security with its backup and restore features.



Example of a RAIC with 3 nodes using FileMaker Server as the backend host

- The Web Server Connector on the web server machine, FileMaker Pro 5.5 Unlimited on multiple machines, and FileMaker Pro 5.5 or FileMaker Pro 5.5 Unlimited on a machine hosting the databases.

In this configuration, you use a dedicated copy of FileMaker Pro or FileMaker Pro Unlimited as the backend host for the databases and use peer-to-peer networking for each of the FileMaker Pro Unlimited guests. (If you use FileMaker Pro Unlimited as the backend host, then you can also use the same machine as a node in the RAIC for the Web Server Connector.)



Using FileMaker Pro or FileMaker Pro Unlimited as the backend host, web users can read and write to the databases

When you set up FileMaker Pro Unlimited on each RAIC machine, you use the network to open the databases and then share each database via the Web. (See “Sharing the database via the Web” on page 3-5 for information.)

Security considerations when using a RAIC

To ensure proper authentication, make sure all nodes of the RAIC have consistent security. For example, you may experience unexpected behavior if you enable security on one machine but not on other machines in the RAIC, or if your FileMaker Pro passwords are different across the nodes of the RAIC.

Where to place database and HTML files

When publishing your FileMaker Pro databases on the Web, place the database files on the host machine running FileMaker Pro Unlimited, FileMaker Server, or FileMaker Pro. Store all your custom web pages that reference the databases in the Web folder inside the FileMaker Pro 5.5 folder. Static HTML and other files (such as image files) should reside on the web server machine, as specified by the web server software. For more information, see “Requirements for web access” on page 3-3.

Requirements for the Web Server Connector

To use the Web Server Connector, you need the following equipment and software.

Requirements on Windows machines

To use the Web Server Connector on Windows systems, you need:

- a hard disk with at least 10MB of free space
- a CD or DVD drive
- Windows 2000 Server
- Microsoft Internet Information Server 5.0

Requirements on Mac OS X Server machines

To use the Web Server Connector on Mac OS X Server you need:

- hard disk with at least 1 MB of free space

- a CD or DVD drive
- Mac OS X Server 10.0.3
- Apache Web Server 1.3.19

Requirements on Linux machines

To use the Web Server Connector on Linux you need:

- a hard disk with at least 42 MB of free space
- a CD or DVD drive
- Red Hat Linux 7.1 i386
- Apache Web Server 1.3.19

Installing on Windows

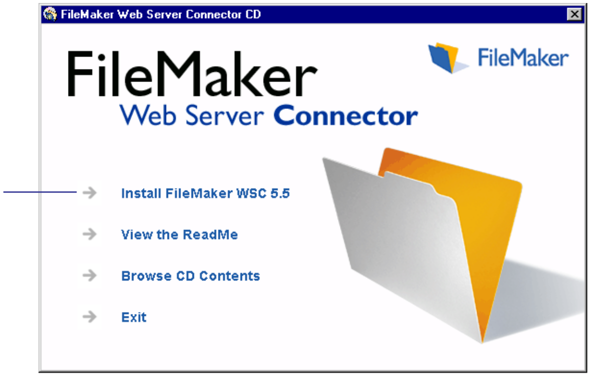
You can use the FileMaker Web Server Connector with Microsoft Internet Information Server 5.0. Before you install the Web Server Connector, you must install the Internet Information Server software on your Windows 2000 machine. (See the documentation for the server software for information.)

Note Installing the Web Server Connector requires you to restart your server. Before you begin the installation process, save your work, exit other open programs, and turn off virus protection utilities.

To install the Web Server Connector:

1. Insert the *FMWSC and Tools* CD into your CD or DVD drive.

2. Click the arrow next to Install FileMaker WSC 5.5 to begin the installation.



Click this arrow

3. In the Welcome screen, click Next.

4. Read the license agreement. If you agree to these terms, select I accept the terms in the license agreement, then click Next.

5. Personalize this copy of the Web Server Connector by typing your name, organization name, and by indicating who can access the Web Server Connector from this computer (only you, or anyone else who uses this computer). Click Next.

6. Choose Complete in the Setup Type screen to install the Web Server Connector, then click Next.

7. Click Install to begin the Web Server Connector file installation.

The Web Server Connector stops installation if it detects another, previously installed servlet engine on your web server. For information about installing the Web Server Connector with other servlet engines, see the Readme.txt file in the Unsupported folder on the *FMWSC and Tools* CD.

8. Click Finish to close the Setup Wizard.

When the installation is finished, you are asked to restart your computer.

9. Click Yes to restart now or No if you want to restart later.

The Web Server Connector automatically starts after you restart your computer. For information on configuring the Web Server Connector with FileMaker Pro Unlimited, see chapter 2, “Administering the Web Server Connector.”

Setting up user names and passwords

Before you can configure the Web Server Connector with FileMaker Pro Unlimited, you must enable the Basic (Clear Text) Password Authentication option in Microsoft Internet Information Server. For instructions on this procedure, see the documentation included with your software.

When using the FileMaker Web Server Connector on Microsoft Internet Information Server, you use a valid user name and password for Windows 2000. Microsoft Internet Information Server instructs Windows 2000 to authenticate the user name and password before passing them on to FileMaker Pro. Do not use high ASCII characters in user names on Windows NT, because these may cause authentication failures.

If you use the FileMaker Pro Web Security Database to control user name and password access, you must establish Windows 2000 accounts with these same user names and passwords. Microsoft Internet Information Server authenticates these names and passwords before passing them on to FileMaker Pro. (For information on using the Web Security Database, see the Web Security.pdf document in the Web Security folder in the FileMaker Pro 5.5 Folder.)

The same user name and password requirements apply when using FileMaker Pro access privileges.

Where files are installed

The files for the Web Server Connector are installed in two main locations: in the FileMaker program folder in a folder named FileMaker WSC 5.5 and in the Scripts folder inside the Inetpub folder:

Contents of the FileMaker WSC 5.5 folder	Description
FileMaker WSC Admin	URL link file
FMWSC.log	The Web Server Connector log file
JSDK License.txt	Java Servlet Development Kit license
License.txt	FileMaker, Inc. license
FMWSCNative.log	The Web Server Connector log file
ReadMe.txt	FileMaker, Inc. Read Me file

Contents of the jre > bin > Security folder	Description
Java.security	Java runtime environment security file

Contents of the lib folder	Description
FMWSC.jar	The FileMaker Web Server Connector servlet
FMWSCResources.jar	Strings and supporting HTML pages for WSC servlet
Servlet.jar	The servlet API

Contents of the Inetpub > Scripts folder	Description
FMWSC_NSAPI.dll	Plug-in API

Uninstalling the Web Server Connector

To uninstall the Web Server Connector from your Windows machine:

1. Stop Microsoft Internet Information Server services.
2. Open the Add/Remove Programs control panel.
3. In the Install/Uninstall tab, select FileMaker WSC 5 from the list.
4. Click Add/Remove.

5. Restart your server.

Note The Uninstall command removes all files that have been installed, but does not remove any files generated by the Web Server Connector. Examples of files not removed include .PROPERTIES files, .LOG files, and ISAPI filter entries, which must all be removed manually.

To remove the ISAPI filter entry:

1. Start the Internet Service Manager application.

2. Right-click on the Internet Information Server icon and choose Properties from the context menu.

3. From the Master Properties menu, select WWW Service, and click Edit.

4. Click the ISAPI Filters tab.

5. Select fmwsc_isapi and click Remove.

6. Click OK, then click OK again.

7. Restart Internet Information Server to reload your remaining filters.

Installing on Mac OS X Server

Note Before you begin the installation process, save your work, quit any open programs, and turn off virus protection utilities. Remember to turn on virus protection utilities again when the installation is complete.

To install the Web Server Connector on Mac OS X Server:

1. If you are not logged in as root, log out and log in as root.

2. Insert the *FMWSC and Tools* CD into your CD or DVD drive.

3. Double-click the file named FileMaker WSC 5.5 Installer.

4. Read the license agreement. If you agree to these terms, click Accept.

You see the Web Server Connector Installer dialog box.

It is recommended that you install the Web Server Connector in the default installation location.

5. Click Install.

6. Click Quit to leave the Installer when the installation is finished.

7. Log out and log in as the user used to administer the machine.

Starting the Web Server Connector

To automatically start up the Web Server Connector, double-click the file named StartFMWSC.command located in the FileMaker_WSC_5.5 folder in the Applications folder.

A terminal window opens telling you that the Web Server Connector is running.

Note If you close the terminal window, the Web Server Connector will stop.

To manually start the Web Server Connector, open a terminal window and enter:

```
# cd /applications/FileMaker_WSC_5.5
# bin/FMWSC_Apache.sh . &
```

To stop the Web Server Connector, you must *kill* the FMWSC_Apache process.

Where files are installed

The files for the Web Server Connector are installed the FileMaker_WSC_5.5 folder:

Contents of the FileMaker_WSC_5.5 folder	Description
FMWSC.log	The Web Server Connector log file (does not appear until the Web Server Connector is first run)
Start_FMWSC.command	Shell script for starting the Web Server Connector

Contents of the bin folder	Description
FMWSC_Apache.sh	Shell script
FMWSC.log	The Web Server Connector log file (does not appear until the Web Server Connector is first run)
FMWSC.properties	The Web Server Connector properties file (does not appear until the Web Server Connector is first run)

Contents of the lib folder	Description
FMWSC_Apache.jar	Apache file for servlet engine
FMWSC.jar	The FileMaker Web Server Connector servlet
FMWSCResouces.jar	Strings and supporting HTML pages for WSC servlet
servlet.jar	The servlet API

Installing on Red Hat Linux

You can use the terminal window or a pseudo terminal (pts) to install the FileMaker Web Server Connector.

Important You must be logged in as the *root* user during installation and when making changes to the Web Server Connector configuration file. Logging in as root gives you complete access to all system resources. Be careful when working as the root user. You could accidentally issue a command that could detrimentally affect your operating system software.

To install the FileMaker Web Server Connector:

1. Insert the *FMWSC and Tools* CD into your CD or DVD drive.

2. Switch to the root user by entering the following command at the shell prompt:

```
$ su -l root
```

For information about accessing the shell prompt, see your operating system documentation.

3. Enter the root password.

The prompt changes to #, indicating that you are now logged in as the root user.

4. If the installation CD does not mount automatically, enter:

```
# mount /dev/cdrom /mnt/cdrom
```

5. Change to the CD directory by entering:

```
# cd /mnt/cdrom
```

6. Change to the Linux directory by entering:

```
# cd linux
```

7. Enter the following command to install the Web Server Connector files on your hard disk:

```
# rpm -ivh fmwsc-5.5-1.i386.rpm
```

Starting the Web Server Connector

1. To start up the Web Server Connector, enter the following:

```
# cd /var/fmwsc/bin
# ./fmwsc_apache.sh path to the www root directory
```

2. Log out as the root user by entering:

```
# exit
```

Where files are installed

File	Description
/var/fmwsc/README	The FileMaker, Inc. Read Me file
/var/fmwsc/fmwsc/FMWSC.log	The Web Server Connector log file (does not appear until the Web Server Connector is first run)
/var/fmwsc/bin/FMWSC_Apache.sh	Shell script
/var/fmwsc/install/mod_fmwsc.so	Shared object used by Apache

File	Description
/var/fmwsc/install/httpd.fmwsc	Appends to the Apache configuration file
/var/fmwsc/lib/FMWSC.jar	The FileMaker Web Server Connector servlet
/var/fmwsc/lib/FMWSC.properties	The Web Server Connector properties file (does not appear until the Web Server Connector is first run)
/var/fmwsc/lib/FMWSCResources.jar	Strings and supporting HTML pages for Web Server Connector servlet
/var/fmwsc/lib/FMWSC_Apache.jar	Apache file for servlet engine
/var/fmwsc/lib/servlet.jar	Servlet API
/var/fmwsc/src/mod_fmwsc.c	Source code for the Web Server Connector Apache module

Uninstalling the Web Server Connector

To remove all files from your hard disk that were installed by the Web Server Connector installer:

1. Switch to the root user by entering the following command at the shell prompt:

```
$ su -l root
```

2. Enter the root password.

3. Enter the following command to remove the Web Server Connector:

```
# rpm -e fmwsc
```

Note It is normal for errors such as error: cannot remove /var/fmwsc/lib - directory not empty to be generated at this point.

4. Log out as the root user by entering:

```
# exit
```

Chapter 2
Administering the Web Server Connector

You administer the FileMaker Web Server Connector from your web browser. FileMaker Pro 5.5 Unlimited and your databases must be set up on your host machine(s) before you can use the Web Server Connector. The first time you use the Web Server Connector, you must configure the administration account. Then you can configure the Web Server Connector with your FileMaker Pro Unlimited host(s) either by host machine or by database.

Setting up FileMaker Pro Unlimited

Each host machine that you're using with the Web Server Connector must have a copy of FileMaker Pro Unlimited installed on it. (For information on using FileMaker Pro or FileMaker Server as a backend host with the Web Server Connector, see "Configuring a RAIC to use with the Web Server Connector" on page 1-1.)

FileMaker Pro Unlimited must be open and running on your host machine(s) and the Web Companion must be enabled. Each of your FileMaker Pro databases must be open and shared via the Web Companion. For information on where to store your other web site files, see "Security considerations when using a RAIC" on page 1-3 and "Requirements for web access" on page 3-3.

To set up FileMaker Pro Unlimited on each host machine:

1. Enable the Web Companion.

This only needs to be done once per machine. For information, see "Enabling the Web Companion" on page 3-3.

2. Configure the Web Companion.

If the host machine is the same as the web server machine, then you must change the Web Companion's TCP/IP port number from 80 to a different number. (The port number 80 is reserved for web servers.) For information, see "Setting Web Companion configuration options" on page 3-3.

3. Open and share each database via the Web Companion.

For information, see "Sharing the database via the Web" on page 3-5.

Configuring the administration account

When you use the Web Server Connector for the first time, you must configure an administration account by entering a user name and password. The Web Server Connector uses the account name and password to control access and prevent unauthorized changes. Because the Web Server Connector runs on top of your web server software, the user name and password must correspond to those of an existing account on your web server. For Windows 2000 Server, this account must also have administrator-level access to the directories your web pages are stored in.

Important The user account must have Password never expires selected in the Computer Managment profile.

To access the Web Server Connector configuration pages:

1. Start your web browser.

2. Enter the IP address or domain name of the web server and a FileMaker CGI request for the FileMaker Web Server Connector:

```
http://IP address/FMPro?config
http://www.domainname.com/FMPro?config
```

If the Web Server Connector is on the same machine as the web server software, then you can enter "localhost" instead of the IP address.

```
http://localhost/FMPro?config
```

For information on using localhost without a network connection, see “Testing your site without a network connection” on page 3-18.

Note On Windows 2000, you can also start the Web Server Connector by choosing Start > Programs > FileMaker WSC - IIS > FileMaker WSC Admin.

3. On the Initial Security page, enter a user name and password in the text boxes.

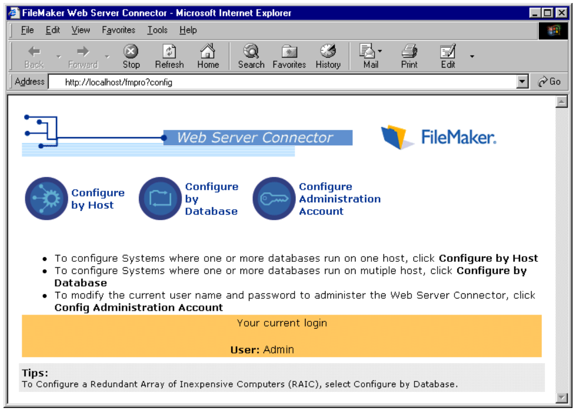
Use the same user name and password as the administrator’s user name and password set up for your web server’s operating system. The user name and password must not contain high ASCII characters. On Windows NT, you enter only a valid user account name.

Note To log in and administer the Web Server Connector you must have cookies enabled.

4. If you make a mistake, you can click Clear Fields and then reenter the user name and password.

5. Click Submit.

The FileMaker Web Server Connector configuration page appears.



Address http://localhost/fmpro?config

- To configure Systems where one or more databases run on one host, click **Configure by Host**
- To configure Systems where one or more databases run on mutiple host, click **Configure by Database**
- To modify the current user name and password to administer the Web Server Connector, click **Config Administration Account**

Your current login

User: Admin

Tips:
To Configure a Redundant Array of Inexpensive Computers (RAIC), select Configure by Database.

FileMaker Web Server Connector configuration page

Important For security purposes, you should quit the web browser when you are done administering the Web Server Connector.

Resetting the administrator account

If you forget or lose your Web Server Connector administrator password, you can reset it by deleting the fmwsc.properties file and restarting the Web Server Connector. This file is re-created the next time the Web Server Connector is launched.

The fmwsc.properties file is located as follows:

Web server	Log file location
Internet Information Server	c:/program files/filemaker/filemaker wsc 5.5/ lib/ fmwsc.properties
Apache on Mac OS X Server	Applications/FileMaker_WSC_5.5/lib/ FMWSC.properties
Apache on Linux	/var/fmwsc/lib/FMWSC.properties

Note On Windows 2000 and Mac OS X Server you must restart the computer to recreate the removed Properties file.

Configuring the Web Server Connector by host machine or by database

You can configure the FileMaker Web Server Connector to send database requests to each FileMaker Pro Unlimited host or database in a list. You can change the configuration at any time by accessing the FileMaker Web Server Connector configuration page.

Accessing the Web Server Connector configuration pages

You can access FileMaker Web Server Connector configuration page by entering one of the following requests in your web browser:

```
http://IP address/FMPro?config
```

```
http://www.domainname.com/FMPro?config
http://localhost/FMPro?config
```

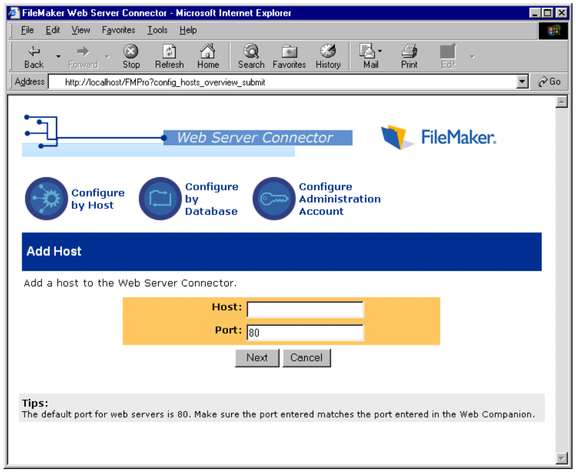
Configuring by host

When you configure the Web Server Connector by host, you specify which databases on each host machine that you want the Web Server Connector to send database requests to. This configuration is useful if you have all of the databases on one or more machines.

To configure the Web Server Connector by host:

1. Click Configure by Host on the FileMaker Web Server Connector configuration page or in the navigation bar of any page.

2. On the Configure by Host web page, click Add Host.


FileMaker Web Server Connector - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit
Address http://localhost/FMPro?config_hosts_overview_submit
Go
Web Server Connector
FileMaker.
Configure by Host
Configure by Database
Configure Administration Account
Next
Cancel

Add Host

Add a host to the Web Server Connector.

Host:

Port: 80

Tips:
The default port for web servers is 80. Make sure the port entered matches the port entered in the Web Companion.

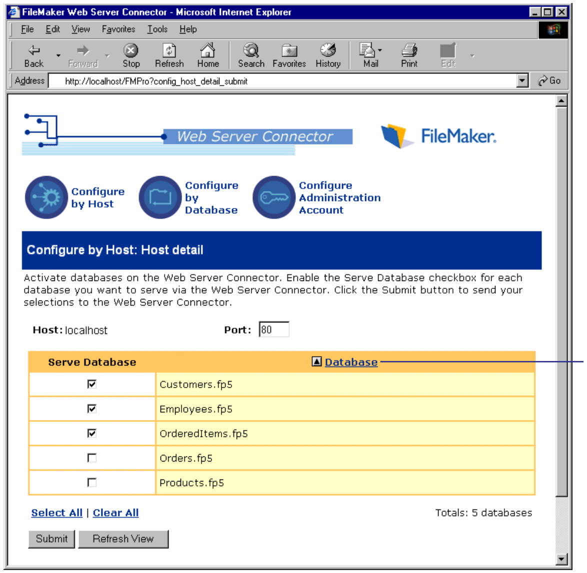
Add Host web page

3. On the Add Host web page, enter the IP address or domain name of the host machine in the Host text box.

4. In the Port text box, enter the port number that is set in the Web Companion Configuration dialog box on the host machine. If the host machine is the same as the web server machine, this port number must be different than Port 80, which is reserved for web servers.

5. Click Next.

A list of all the databases that are open and shared via the Web Companion on the host machine appears in the Host Detail web page.


FileMaker Web Server Connector - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit
Address http://localhost/FMPro?config_host_detail_submit
Go
Web Server Connector
FileMaker.
Configure by Host
Configure by Database
Configure Administration Account
Select All | Clear All
Submit
Refresh View

Configure by Host: Host detail

Activate databases on the Web Server Connector. Enable the Serve Database checkbox for each database you want to serve via the Web Server Connector. Click the Submit button to send your selections to the Web Server Connector.

Host: localhost Port: 80

Serve Database	Database
☑	Customers.fp5
☑	Employees.fp5
☑	OrderedItems.fp5
□	Orders.fp5
□	Products.fp5

Totals: 5 databases

Click to sort the list of databases

Host Detail web page

6. Click the Serve Database checkboxes to enable each database in the list that you want the Web Server Connector to send requests to. You can also click Select All to enable all the databases in the list.

To sort the list of databases, click Database at the top of the list.

7. Click Submit.

8. To redisplay the complete list of databases on the host machine, click Refresh View.

9. Repeat step 3 through step 7 for each host machine.

To remove a database from the list:

1. Select the database in the list.

2. Deselect the Serve Database checkbox.

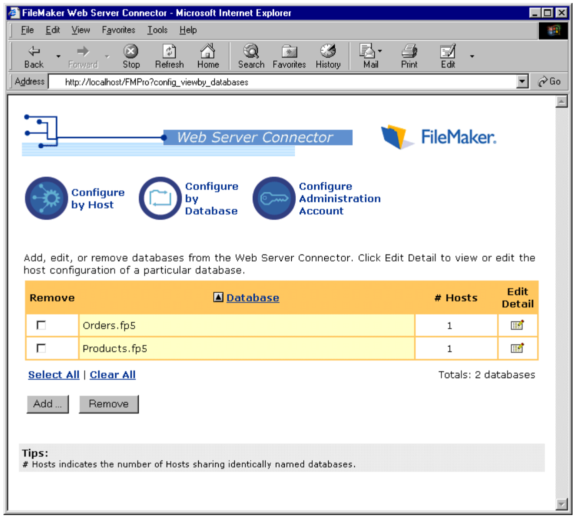
3. Click Submit.

Configuring by database

When you configure the Web Server Connector by database, you set up a list of IP addresses or domain names of the machines that are hosting the database(s). The Web Server Connector will send database requests in a round-robin fashion to the first available host in the list. Use this configuration if you are configuring a single database on multiple hosts.

To configure the Web Server Connector by database:

1. Click Configure by Database on the FileMaker Web Server Connector configuration page or in the navigation bar of any page.



http://localhost/FMPro?config_viewby_databases

Add, edit, or remove databases from the Web Server Connector. Click Edit Detail to view or edit the host configuration of a particular database.

Remove	Database	# Hosts	Edit Detail
☐	Orders.fp5	1	
☐	Products.fp5	1	

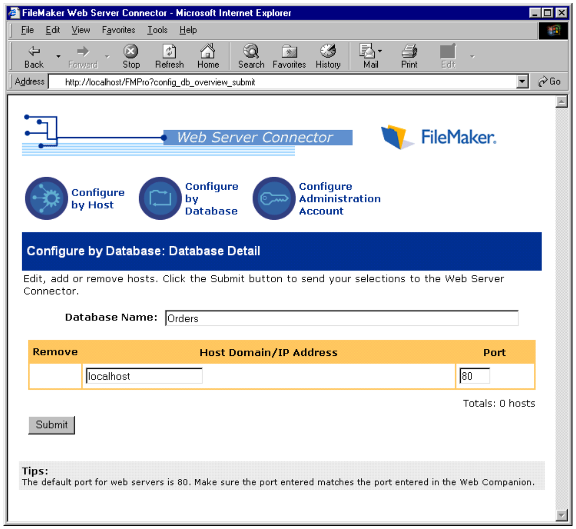
Totals: 2 databases

Tips:
Hosts indicates the number of Hosts sharing identically named databases.

Configure by Database web page

2. On the Configure by Database web page, click Add Database.

The Database Detail page appears.



http://localhost/FMPro?config_db_overview_submit

Configure by Database: Database Detail

Edit, add or remove hosts. Click the Submit button to send your selections to the Web Server Connector.

Database Name: Orders

Remove	Host Domain/IP Address	Port
	localhost	80

Totals: 0 hosts

Tips:
The default port for web servers is 80. Make sure the port entered matches the port entered in the Web Companion.

Database Detail web page

3. In the Database Name text box, enter the name of the database that you want to serve.

Note You can enter the name of a database even if it is not currently hosted by the web server, as long as it is not made available to web users until it’s hosted.

4. Enter the IP address or domain name of the host machine in the text box.

If the host machine is the same as the web server machine, you can enter `localhost` for the IP address.

5. In the Port text box, enter the port number that is set in the Web Companion Configuration dialog box on the host machine.

If the host machine is the same as the web server machine, this port number must be different than Port 80, which is reserved for web servers.

6. Click Submit.

7. Repeat steps 2 through 6 for each database and host machine you want to add to the configuration.

To remove a database from the list:

1. Click Configure by Database on the FileMaker Web Server Connector configuration page or in the navigation bar of any page.

2. Locate the database you want to remove.

3. Select the Remove checkbox next to the database name.

4. Click Remove.

Using the Web Server Connector log files

The Web Server Connector and the Web Companion each generate log files that are useful for troubleshooting purposes. The Web Server Connector log files can help you diagnose problems with the Web Server Connector on your web server, and the Web Companion log files can help you isolate problems on individual database host machines.

The Web Server Connector generates two log files: FMWSC.log and FMWSCNative.log. These files list the date and time of each server start and close event, as well as all errors reported by the servlet engine and the relay servlet.

For information about the log files generated by the Web Companion, see “Web Companion support for Internet media types” on page 3-13.

The Web Server Connector log files are located on the following web servers:

Web server	Log file location
Internet Information Server	c:/program files/filemaker/filemaker wsc 5.5/ FMWSC.log c:/program files/filemaker/filemaker wsc 5.5/ FMWSCNative.log
Apache on Mac OS X Server	Applications/FileMaker_WSC_5.5/ FMWSC.log
Apache on Linux	/var/fmwsc/FMWSC.log

Accessing hosted databases

After the databases are hosted by FileMaker Pro 5.5 Unlimited, and the Web Server Connector is installed and configured, you are ready to access your FileMaker Pro databases over the Internet or intranet.

Connecting to FileMaker Pro custom web pages

You can use your web browser to view the HTML page or index page of the databases you want to access:

`http://`*`web server IP address or domain name/`*
`mysolution.htm`

Use the format 123.456.789.123/mysolution.htm or www.yourdomainname.com/mysolution.htm for *web server IP address or domain name/mysolution.htm.*

For more information see “Creating a custom home page” on page 3-5.

Connecting to FileMaker Pro instant web pages

Note Instant Web Publishing is not supported under the Web Server Connector in FileMaker Pro 5.5 Unlimited.

1. Point your web browser to the IP address or domain name of the web server.

`http://`*`web server IP address or domain name/`*

Use the format 123.456.789.123 or www.yourdomainname.com for *web server IP address or domain name.*

You see the FileMaker Pro default page and a list of links to the currently shared databases.

2. From the list, click the database you want to access.

For more information see “Creating a custom home page for Instant Web Publishing” on page 3-6 and “Creating a custom web site using a database layout” on page 3-8

Note To use Instant Web Publishing, you must dedicate the host machine(s) for it alone (you can’t also use custom web publishing from these machines) and configure the Web Companion to accept Instant Web Publishing requests. See “Setting Web Companion configuration options” on page 3-3 for information.

Chapter 3
Publishing your database on the Web

The FileMaker Pro Web Companion plug-in makes it possible for you to publish your database on the Internet or an intranet in several different ways, giving you more choices and control over the design and functionality of your web pages.

You can publish your database with:

- custom web publishing using XML
- FileMaker Pro database-aware Java applets using the FileMaker JDBC Driver or the proprietary FileMaker Java Class Library
- custom web publishing using CDML
- FileMaker Pro Instant Web Publishing
- a custom home page using Instant Web Publishing
- custom web publishing using a database layout
- static web publishing (exporting data into an HTML table)

When you serve your databases and your instant or custom web pages via the Web Companion, users can access your databases from their web browsers using a simple URL.

This chapter gives general information on setting up and using the Web Companion for creating custom web pages.

Note Instant Web Publishing is not supported under the Web Server Connector in FileMaker Pro 5.5 Unlimited.

Types of web publishing

Custom web publishing with XML

You can use the Web Companion to generate data from your FileMaker Pro databases into Extensible Markup Language (XML) documents. With these XML documents, you can (for example) use JavaScript and the W3C Document Object Model to dynamically manipulate data after it has been downloaded from your database. Many of the actions (such as searching) can be performed without the need to reconnect to the database, making the web user's interaction with the database happen much faster.

For information on custom publishing your database with XML, see chapter 5, "Using FileMaker Pro XML to deliver your data."

For a list of XML resources and examples, and an overview of what XML is and how it can be used with FileMaker Pro, see "XML and FileMaker 5—a Technology Preview" on the FileMaker support pages at www.filemaker.com. As a shortcut, double-click FileMaker on the Web (included on the *FileMaker Pro* CD).

Custom web publishing with JDBC

The *FMWSC and Tools* CD provides a JDBC (Java database connectivity) API-compatible driver that allows you to create FileMaker Pro database-aware Java applets for your web site using any Rapid Application Development (RAD) tool. This is an improvement from using the proprietary FileMaker Java classes that are not recognizable by RAD tools.

With a FileMaker Pro database-aware Java applet, you can make your databases function more like they are being used in FileMaker Pro rather than in a web browser.

For information, see chapter 6, "Using Java and JDBC to deliver your data."

For a list of JDBC resources, see the product support pages on the FileMaker, Inc. web site at www.filemaker.com. As a shortcut, double-click FileMaker on the Web (included on the *Filemaker Pro* CD).

Custom web publishing with CDML

The FileMaker Pro Web Companion lets you publish your database with custom web pages using a proprietary markup language called CDML. Included with FileMaker Pro 5.5 Unlimited are all the tools, templates, and examples you need to create your own custom web pages using CDML. For information, see chapter 4, "Custom web publishing using CDML."

Instant Web Publishing

When you use FileMaker Pro Instant Web Publishing, the Web Companion generates predesigned HTML forms with HREF links to your database. For general information on web publishing and using the Instant Web Publishing feature, see chapter 14, "Publishing databases on the Web," in the *FileMaker Pro User's Guide* or see FileMaker Pro Help.

Note Instant Web Publishing is not supported under the Web Server Connector in FileMaker Pro 5.5 Unlimited.

Other ways to create custom web sites for your data

If you want, you can create a custom home page to go with your instant web pages instead of using the built-in FileMaker Pro Instant Web Publishing home page. See "Creating a custom home page" on page 3-5, "Creating a custom home page for Instant Web Publishing" on page 3-6, and "Creating a custom web site using a database layout" on page 3-8.

Static web publishing with HTML

You can also publish your data on static web pages if you don't need dynamic web access to your database. See "Exporting data to a static HTML page" on page 3-16.

Using the FileMaker Pro Web Companion

The Web Companion is a plug-in that acts as a Common Gateway Interface (CGI) application for handling interactions between FileMaker Pro and your web browser. The Web Companion also functions as a web server by providing static files (such as HTML pages and images) to the web browser.

Web users access your database either by accessing the IP address of the computer running FileMaker Pro Unlimited with their browser (which takes them to the home page) or by clicking an HREF link that contains a specific CGI request for FileMaker Pro. The Web Companion then sends via HTTP (Hypertext Transfer Protocol) either the default home page or the web page specified in the FileMaker CGI request.

The Web Companion in FileMaker Pro Unlimited can serve your databases to an unlimited number of IP addresses at any time. (The Web Companion in FileMaker Pro can only serve to a maximum of ten IP addresses in a 12-hour period, as indicated by the IP Guest Limit of 10 in the Web Companion Configuration dialog box.)

If desired, you can set up your computer for testing without a constant connection to the Internet or an intranet. For information, see "Testing your site without a network connection" on page 3-18.

For general information on the Web Companion and about connecting to the Internet or an intranet, see chapter 14, "Publishing databases on the Web" in the *FileMaker Pro User's Guide*.

Requirements for web access

The host computer must have a copy of FileMaker Pro Unlimited serving the databases on the Web (preferably with a full-time, constant connection to the Internet or your intranet). The Web Companion must be enabled in FileMaker Pro Unlimited.

In addition, your site folders and web pages must be located inside the Web folder (inside the FileMaker Pro 5.5 application folder) in order for the Web Companion to serve them on the Web. However, your databases do not have to be inside the Web folder. They only need to be open in FileMaker Pro Unlimited and shared via the Web Companion. For security purposes, it's a good idea not to have your databases or other sensitive documents in the Web folder.

Note You can keep your site folders, web pages, and databases in a different folder anywhere on your hard drive. To do this, replace the Web folder inside the FileMaker Pro 5.5 folder with a shortcut/alias named "Web".

For information and tips on providing security for your databases on the Web, see the WebSecurity.pdf document in the FileMaker Pro 5.5 Web Security folder.

Enabling the Web Companion

You only need to enable the FileMaker Pro Web Companion plug-in once. FileMaker Pro Unlimited will attempt to connect to a network in order to enable the Web Companion — if you do not have a network connection but want to enable the Web Companion anyway, see "Testing your site without a network connection" on page 3-18.

Mac OS X To enable the Web Companion on Mac OS X machines, see appendix C, "Enabling the FileMaker Pro Web Companion in Mac OS X."

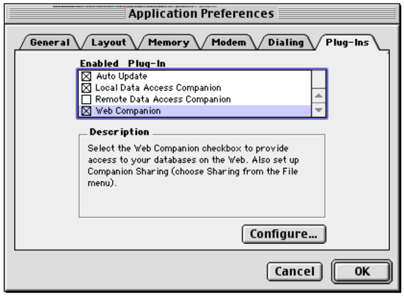
To enable the Web Companion:

1. In FileMaker Pro, choose Edit menu > Preferences > Application.

2. In the Application Preferences dialog box, click the Plug-Ins tab.

If Web Companion doesn't appear in the list in the Application Preferences dialog box, you must install the FileMaker Pro Web Support component. See the sections on custom installation in the *FileMaker Pro 5.5 Getting Started Guide* for information.

3. Select the Web Companion checkbox to enable the Web Companion plug-in.



Application Preferences

Enabled Plug-In

☒ Auto Update
☒ Local Data Access Companion
☐ Remote Data Access Companion
☒ Web Companion

Description

Select the Web Companion checkbox to provide access to your databases on the Web. Also set up Companion Sharing (choose Sharing from the File menu).

Enable the Web Companion in the Application Preferences dialog box

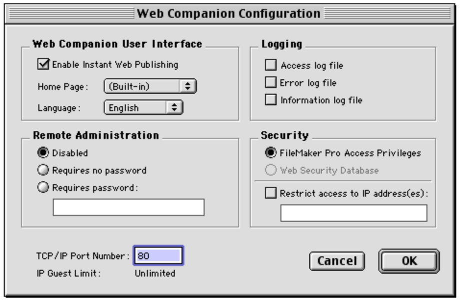
4. Select Web Companion and click Configure to set configuration options, or click OK.

Important FileMaker does not recommend using Mac OS X machines in a RAIC configuration. If you must, the Mac OS X machines should be set to port 1024 or higher. For information about changing ports, see "Configuring the Web Companion for use with ports 1024 and higher" on page C-2.

Setting Web Companion configuration options

After you've enabled the Web Companion, follow these steps to select various configuration options:

1. On the Plug-Ins tab in the Application Preferences dialog box, select Web Companion and click Configure.



Web Companion Configuration

Web Companion User Interface

☑ Enable Instant Web Publishing

Home Page: (Built-in)

Language: English

Logging

□ Access log file

□ Error log file

□ Information log file

Remote Administration

● Disabled

○ Requires no password

○ Requires password:

Security

● FileMaker Pro Access Privileges

○ Web Security Database

□ Restrict access to IP address(es):

TCP/IP Port Number: 80

IP Guest Limit: Unlimited

Example of the Web Companion configured for Instant Web Publishing

2. In the Web Companion Configuration dialog box, choose an HTML file from the Home Page list so the Web Companion will automatically display it when web users enter the IP address.

The [Built-in] option displays the “FileMaker Pro Instant Web Publishing” home page by default. All other HTML files that are located in the root level of the Web folder appear in this list.

See “Creating a custom home page” on page 3-5 for more information.

3. For custom web publishing, deselect Enable Instant Web Publishing.

4. If desired, choose a language from the Language pop-up menu for using localized texts in Instant Web Publishing navigation.

This choice will not affect the language of your data.

Note The Language setting can also be used with the [FMP-CurrentAction], [FMP-FindOpItem], or [FMP-SortOrderItem] CDML replacement tags in your custom web pages. See “Using an encoding parameter with a CDML replacement tag” on page 4-14 for information.

5. If you want, select one or more Logging options: Access Log File, Error Log File, and Information Log File.

For information, see “Web Companion support for Internet media types” on page 3-13.

6. Select a Remote Administration option.

If you want to remotely access the Web folder from a different computer, for example, to upload or download files using HTTP Put and Get commands or to change settings in the Web Security Database, select Requires Password and enter a password in the box. (If it doesn’t matter who has access to the Web folder and everything inside it, select Requires no password.)

For more information, see “Opening password-protected databases remotely” on page 3-18.

7. Select a Security option.

FileMaker Pro Access Privileges is selected by default. For general information about setting access privileges in FileMaker Pro, see chapter 9, “Protecting databases with passwords and groups,” in the *FileMaker Pro User’s Guide.*

In FileMaker Pro 5.5, you can now secure data on a record-by-record basis using access privileges. For more information, see the Web Security.pdf file in the FileMaker Pro 5.5 Web Security folder or see FileMaker Pro Help.

The Web Security Database option is not available when the Web Security.fp5 database is not open. For information on the Web Security database, see the Web Security.pdf file.

8. If desired, select Restrict access to IP address(es) and type the IP addresses of the computers that are allowed to access the Web folder, the web pages served by the Web Companion, and your databases.

You can enter multiple IP addresses separated by commas and use an asterisk as a wildcard for all addresses beginning with the specified numbers. For example, 1.2.3.4, 5.6.7.8, 3.5.* indicates two IP addresses and all addresses that begin with “3.5.”

A computer’s IP address is determined by the network administrator (for an intranet) or an Internet service provider (ISP) account.

9. Specify a TCP/IP port number.

By default, web browsers use the TCP/IP port number 80 to communicate with the web server. If that port is in use, you can use any number between 1024 and 65535 or the port number 591, which is registered by FileMaker, Inc. with the Internet Assigned Numbers Authority (IANA) for use with the FileMaker Pro Web Companion.

Mac OS X To use port numbers below 1024 on Mac OS X machines, you'll need your Mac OS X administrator name and password. See appendix C, "Enabling the FileMaker Pro Web Companion in Mac OS X."

10. Click OK to close the Web Companion Configuration dialog box.

11. Click OK again to close the Application Preferences dialog box.

Sharing the database via the Web

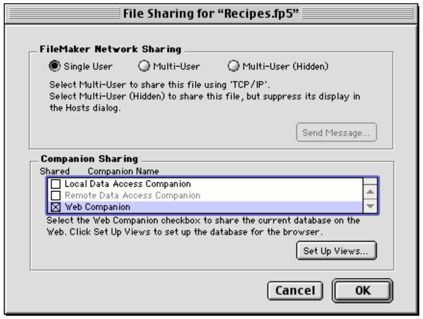
Each database that you're publishing on the Web must be open and shared via the Web Companion.

To share a database on the Web:

1. In FileMaker Pro, choose File menu > Open and open the database.

2. Choose File menu > Sharing.

3. Select the Web Companion checkbox.



File Sharing for "Recipes.fp5"

FileMaker Network Sharing

● Single User ○ Multi-User ○ Multi-User (Hidden)

Select Multi-User to share this file using 'TCP/IP'.
Select Multi-User (Hidden) to share this file, but suppress its display in the Hosts dialog.

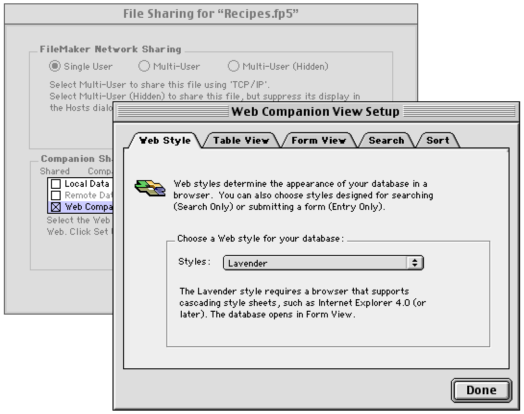
Companion Sharing

Shared	Companion Name
□	Local Data Access Companion
□	Remote Data Access Companion
☒	Web Companion

Select the Web Companion checkbox to share the current database on the Web. Click Set Up Views to set up the database for the browser.

If the Web Companion option is unavailable (dimmed), you need to enable the Web Companion. See "Enabling the Web Companion" on page 3-3.

4. If you're using Instant Web Publishing for this database, click Set Up Views and choose a web style and layouts for each view (instant web page).



Web Companion View Setup

Web styles determine the appearance of your database in a browser. You can also choose styles designed for searching (Search Only) or submitting a form (Entry Only).

Choose a Web style for your database:

Styles: Lavender

The Lavender style requires a browser that supports cascading style sheets, such as Internet Explorer 4.0 (or later). The database opens in Form View.

For Instant Web Publishing, choose a web style and layouts for each view (instant web page)

5. Click Done to close the Web Companion View Setup dialog box.

6. Click OK to close the File Sharing dialog box.

Creating a custom home page

You can set the Web Companion to open a custom home page by default rather than the built-in home page (the FileMaker Pro Instant Web Publishing home page) — whether you're using Instant Web Publishing or custom web publishing with CDML or XML.

When web users enter the IP address of the host computer in their web browsers, the Web Companion will serve either the built-in home page used for Instant Web Publishing, any web page named “default.htm,” “default.html,” “index.htm,” or “index.html” that is located in the root level of the Web folder or in a site folder within it, or the custom home page you specify in the Web Companion Configuration dialog box (which must be located in the root level of the Web folder).

If you’re hosting multiple sites each with its own home page, you can have each home page named “default.htm” or “index.htm” inside each site folder within the Web folder, and the Web Companion will display them when web users enter the name of the site folder after the IP address.

```
http://17.17.17.17/guest_book
```

If you want, create a link to the IP address of the site folder—otherwise, let users know so they can type it in their web browsers.

Note You can enter “localhost” instead of the IP address when FileMaker Pro and all the files are on your computer. See “Testing your site without a network connection” on page 3-18.

```
http://localhost
http://localhost/guest_book
```

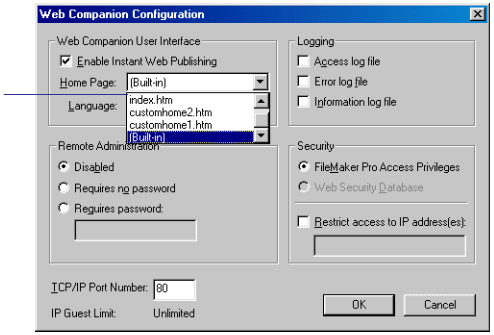
Specifying a custom home page as the default

To specify the custom home page as the default in the Web Companion Configuration dialog box:

1. Make sure the custom home page is located in the root level of the Web folder.

The custom home page can be named anything but must be in HTML format (with the .htm or .html filename extension).

2. In the Web Companion Configuration dialog box, choose your custom home page from the Home Page list.



Any .htm or .html file located in the root level of the Web folder appears in this list

Web Companion Configuration

Web Companion User Interface

☑ Enable Instant Web Publishing

Home Page: (Built-in)

Language:

index.htm
customhome2.htm
customhome1.htm
(Built-in)

Logging

☐ Access log file
☐ Error log file
☐ Information log file

Remote Administration

● Disabled
○ Requires no password
○ Requires password:

Security

● FileMaker Pro Access Privileges
○ Web Security Database

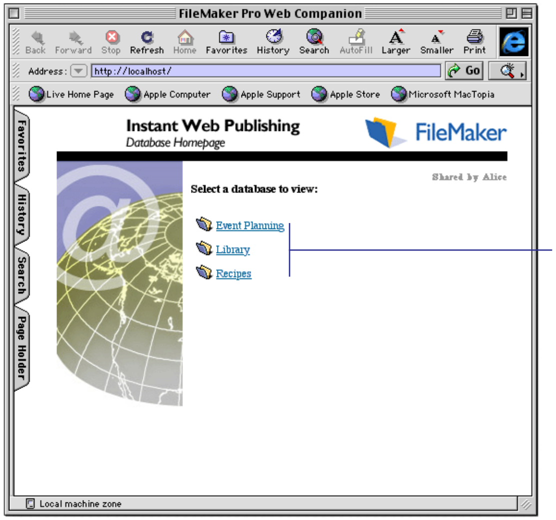
☐ Restrict access to IP address(es):

TCP/IP Port Number: 80

IP Guest Limit: Unlimited

Creating a custom home page for Instant Web Publishing

The simplest and quickest way to publish your FileMaker database on the Web is to let FileMaker Pro Instant Web Publishing design your web pages for you. You can create your own web page to replace the built-in FileMaker Pro Instant Web Publishing home page and still use the instant web pages generated by the Web Companion. Your new custom home page can contain anything else you want to include for your web site, such as web graphics, movies, and Macromedia Flash animations.



Databases shared by the Web Companion

Example of the built-in Instant Web Publishing home page

The built-in home page for Instant Web Publishing displays a list of all the open FileMaker Pro databases that are shared via the Web Companion. Each database in the list is linked to a FileMaker CGI request for displaying the database records in different views (instant web pages), such as Form View.

About the FileMaker WebPortal object

With the enhanced Web Companion in FileMaker Pro 5.5 and FileMaker Pro 5.5 Unlimited, you can now access the elements of the Instant Web Publishing home page (such as database names or the URL to access a Form View of a database) as separate JavaScript objects and extract data from them to build your own custom home page.

To access elements of the Instant Web Publishing home page, you need the following FileMaker CGI request:

```
<SCRIPT LANGUAGE="JavaScript" SRC="FMPro?-webportal">
</SCRIPT>
```

Note If you are using a Netscape web browser, you must specify JavaScript 1.4.

This HTML statement places the JavaScript object called WebPortal inside a window object in your web page. The window.webPortal object contains the following subobjects:

webPortal.databases = Array of <databaseObject>
webPortal.userName = Name of current FileMaker Pro user

Each <databaseObject> in the array contains the following:

databaseName	The string displayed for the link in the Instant Web Publishing home page
defaultURL	The default URL for opening the database in the browser based on settings made in the Web Companion View Setup dialog box or the Document Preferences dialog box
formViewURL	The URL for opening the database in the Form View instant web page
tableViewURL	The URL for opening the database in the Table View instant web page
searchViewURL	The URL for opening the database in the Search View instant web page
newViewURL	The URL for opening the database in the New Record View instant web page

Note A URL for the default Form View for a database can be used with FileMaker WebPortal objects.

A Custom Web Portal example is included in the Custom Workgroup Portal folder on the *FMWSC and Tools* CD.

Overview of setting up a custom home page for Instant Web Publishing

To create a custom home page using JavaScript:

1. Create an HTML file for your web page using a text editor or HTML editing program.

2. Include a FileMaker CGI request for the FileMaker WebPortal object:

```
<SCRIPT LANGUAGE="JavaScript" SRC="FMPro?-webportal">
</SCRIPT>
```

3. As desired, use scripting to write HTML and text to the document.

4. Save the file with the .htm or .html filename extension and place it in the root level of the Web folder (inside the FileMaker Pro 5.5 folder).

5. In the Web Companion Configuration dialog box, select Enable Instant Web Publishing, choose your custom web page from the Home Page list, and click OK.

Note Instant Web Publishing is not supported under the Web Server Connector in FileMaker Pro 5.5 Unlimited.

Creating a custom web site using a database layout

With new features in FileMaker Pro 5.5 and FileMaker Pro 5.5 Unlimited, you can now design your own page layouts for Instant Web Publishing in Layout mode, and then display the layouts in the web browser.

First, you create a startup script to hide the Instant Web Publishing interface. Then you create buttons in the layouts to navigate the web site and perform database functions. To bypass the Instant Web Publishing home page, you use a custom home page that contains a redirect statement for opening the database layout in a particular instant web page.

Overview of using a database layout as the Instant Web Publishing home page

To create a custom home page that uses a database layout for the interface:

1. In Layout mode in FileMaker Pro, design the layout for your custom home page. Create script buttons for every type of interaction you want the web user to be able to do on this page. (See "Using script buttons in Instant Web Publishing" on page 3-9.)

2. Create a startup script that hides the Instant Web Publishing interface. (See "Suppressing the Instant Web Publishing interface" on page 3-11.)

3. Enable the Web Companion and configure it for Instant Web Publishing.

4. Set up file sharing for the database through the Web Companion (see "Sharing the database via the Web" on page 3-5), and click Set Up Views in the File Sharing dialog box.

5. In the Web Companion View Setup dialog box, choose either Soft Gray, Lavender, Wheat, or Blue and Gold 1 for the web style. Then specify the layout you want used for each view (instant web page).

6. Choose Edit > Preferences > Document and set up FileMaker Pro to switch to the layout you created in step 1 and perform the startup script you created in step 2 when the database opens.

When the database is opened in the web browser, the layout that you select for the Switch to Layout option in the Document Preferences dialog box overrides all layouts selected in the Web Companion View Setup dialog box for file sharing.

7. Create a web page with a redirect statement that bypasses the built-in Instant Web Publishing home page and displays your custom layout in the browser. (See "Bypassing the Instant Web Publishing home page" on page 3-12.)

8. Configure the Web Companion to open the web page as the default home page. (See "Specifying a custom home page as the default" on page 3-6.)

Now when you enter your computer's IP address or "localhost" in the web browser, the Web Companion will display the database layout for your custom home page in the browser window.

Using script buttons in Instant Web Publishing

You can provide special script buttons in your FileMaker Pro layout to work with Instant Web Publishing. When web users click on a button in the browser, the script's URL is sent to the Web Companion as a FileMaker CGI request.

If the script is a single script step, an onClick JavaScript event handler is executed and a URL is generated containing the current state information in the browser and information from the script step extracted by the Web Companion.

If the script contains multiple script steps, state information from the first three supported steps is extracted to construct a JavaScript state object (scriptState) that encapsulates the result of the executed script. The resulting information is passed to the JavaScript runtime application (Instant Web Publishing), which interprets the state object, builds the resulting URL, and sends the CGI request to the Web Companion.

Note For information about FileMaker CGI requests made in custom web publishing, see "Generating FileMaker Pro CGI requests using CDML" on page 4-3 and "Generating FileMaker Pro CGI requests for an XML document" on page 5-8.

Requirements for Instant Web Publishing buttons

A button that you're using in a layout for Instant Web Publishing may have a single valid script step attached to it or a script containing 1 to 3 valid script steps.

If you're using multiple text and graphic objects for a button, the script or script step must be attached to the topmost object in the group. Create the text and graphic elements first, group them, and then attach the script to the group.

Single script steps supported for Instant Web Publishing

In FileMaker Pro 5.5 and FileMaker Pro 5.5 Unlimited, the Web Companion supports the following single script steps for buttons used in Instant Web Publishing layouts.

FileMaker script step	CGI request	Description
Open [<Document name>]	Open database in browser window	Equivalent to opening the database from the Instant Web Publishing home page. The database must be specified as a script parameter, and it must be open in FileMaker Pro.
Open URL [<url>]	Set window location to the specified URL	Use this with a text field or calculation field with a text result to construct target URLs. The URL must be complete (e.g. include http://) and can go to another web site or contain a FileMaker CGI request.
Go to Layout [<Layout Name>]	Go to specified layout	This will not affect other current parameters for location. The generated URL link is based on the default URL of the database plus any settings made for startup script and layout, and specified record in a relationship. *Also supported in a multi-step button.*

FileMaker script step	CGI request	Description
Go to Related Record [<Relationship Name>]	Go to specified record in a related database	The related database must be open and shared via the Web Companion, and the specified record must be available when the script is performed.
Go to Record/Request/Page [n]	Go to record number	The record number can be specified by a constant, by a field value, or from a JavaScript prompt. *Also supported in a multi-step button.*
Go to Record/Request/Page [First]	Go to first record	*Also supported in a multi-step button.*
Go to Record/Request/Page [Last]	Go to last record	*Also supported in a multi-step button.*
Go to Record/Request/Page [Previous]	Go to previous record	*Also supported in a multi-step button.*
Go to Record/Request/Page [Next]	Go to next record	*Also supported in a multi-step button.*
Go to Field [ ]	Go to Edit Record view	Displays the specified field in the Edit Record page (without a blinking insertion point). *Also supported in a multi-step button.*
New Record/Request [ ]	Go to New Record view	*Also supported in a multi-step button.*

FileMaker script step	CGI request	Description
Enter Browse Mode [ ]	Go to Form View	Allows the web user to go from an Edit Record, Search, or New Record page to Form View without submitting a CGI request. *Also supported in a multi-step button.*
Enter Find Mode [ ]	Go to Search page	*Also supported in a multi-step button.*
Show All Records	Find all records	*Also supported in a multi-step button.*
Perform Find	Submit -find request	This does not restore a -find request that has been saved with the script. *Also supported in a multi-step button.*
Exit Record	Submit form	This submits -edit record, -new record, and -find requests. *Also supported in a multi-step button.*
Sort [ ]	Go to Sort page	*Also supported in a multi-step button.*
Delete Record/Request [ ]	Delete record with confirmation alert	
View As [View As Table]	View current layout in CSS table	
View As [View As Form]	View current layout in CSS form	
View As [View As List]	View current layout in CSS table	This is the same as the View As [View As Table] script step

FileMaker script step	CGI request	Description
View As [Cycle]	Cycle between Form and Table views	
Open Help	Open HTML help	This opens the built-in help for Instant Web Publishing, called “FileMaker Pro Web Companion Help” in a new browser window.

Note When you use the Go to Related Record script step with Instant Web Publishing, the sort order of the related database is based on the Sort View settings in the Web Companion View Setup dialog box (not on the sort order specified by the relationship). The found set of records is determined by the relationship — only related records are in the current found set when the script is performed.

Support for scripts with multiple script steps

The Web Companion supports 1 to 3 script steps in a script button used for Instant Web Publishing. (Any steps after 3 valid script steps are ignored.) The script must include a change of mode, layout, or current record. The script may also include the submission of a form containing either a -find request or an edited record. The following script steps can be used in multi-step scripts:

Go to Record
Go to Layout
Go to Field
Sort
Enter Find Mode
Enter Browse Mode
New Record
Show All Records
Exit Record
Perform Find

If the first script step in the script is not one of these supported script steps, then it is handled as a single script step. If it is not supported for Instant Web Publishing, then the script is not generated for the button. If a script contains both supported and unsupported steps, then parsing of the script will cease as soon as the first unsupported step is encountered.

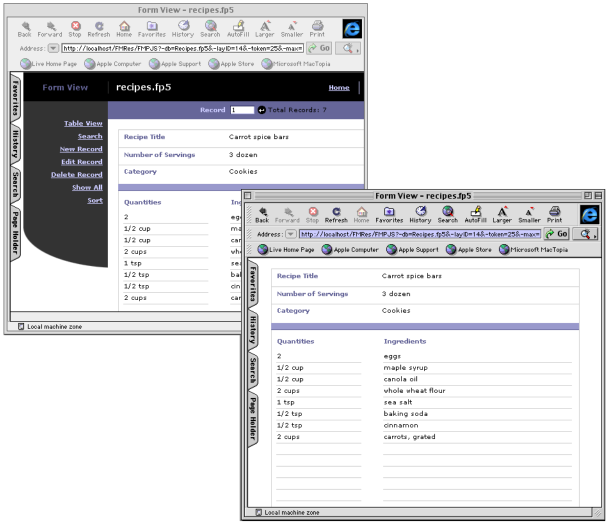
Suppressing the Instant Web Publishing interface

You can use a startup script to suppress the automatic page layouts and navigation controls of Instant Web Publishing in the browser. When web users click a link on the Instant Web Publishing home page, your database layout appears instead of the built-in layout of the instant web page. (To display a database layout instead of the Instant Web Publishing home page, see “Bypassing the Instant Web Publishing home page” on page 3-12.)

When you hide the Instant Web Publishing interface, you must specify one of the following web styles that use cascading style sheets:

- Soft Gray
- Lavender
- Wheat
- Blue and Gold 1

The other web styles don’t work with hiding the Instant Web Publishing interface. For information about web styles, see “Choosing a web style” in chapter 14 of the *FileMaker Pro User’s Guide* or see FileMaker Pro Help.



http://localhost/FMRes/FMPJS?-db=Recipes.fp5&-layID=14&-token=25&-max=

Record 1 Total Records: 7

Recipe Title	Carrot spice bars
Number of Servings	3 dozen
Category	Cookies

Quantities	Ingredients
2	eggs
1/2 cup	maple syrup
1/2 cup	canola oil
2 cups	whole wheat flour
1 tsp	sea salt
1/2 tsp	baking soda
1/2 tsp	cinnamon
2 cups	carrots, grated

Example of a web page before and after the Instant Web Publishing interface is hidden

You only need the Toggle Status Area [Hide] script step in your startup script to hide the Instant Web Publishing interface. In addition, you can combine the Toggle Status Area [Hide] script step with one of the following script steps in the startup script:

- Enter Browse Mode: Form View
- Enter Find Mode: Search page
- New Record: New Record View
- View As [View as Table]: Table View

The Freeze Window, Set User Capture, and Refresh Window script steps can appear before the supported steps, but they will be ignored.

To hide the Instant Web Publishing interface:

1. Choose Scripts > ScriptMaker and type a name for the new script in the Script Name text box. Then click Create.

2. In the Script Definition dialog box, click Clear All, and select Toggle Status Area. Choose Hide from the Specify pop-up menu to add the parameter to the script step. Then click OK.

3. Click Done to close the Define Scripts dialog box.

4. Choose Edit > Preferences > Document.

5. In the Document Preferences dialog box, select the checkbox for Perform script when opening the database, and choose the script you named in step 1 from the pop-up menu.

6. Click OK.

For more information, see “Defining scripts” in chapter 10 and “Setting document preferences” in appendix A of the *FileMaker Pro User’s Guide* or see FileMaker Pro Help.

Bypassing the Instant Web Publishing home page

You can bypass the built-in Instant Web Publishing home page so that the database layout you’ve created appears as the default home page in the web browser. You do this by writing a redirect statement in an HTML file that includes a FileMaker CGI request and then designating the file as the default home page in FileMaker Pro.

For the FileMaker CGI request, you’ll need to know the URL of the view (instant web page) that you want the database layout to appear in. You can get this from the browser window by displaying the database in the Instant Web Publishing home page and moving the cursor over the link or clicking the link to go to the view.

To bypass the Instant Web Publishing home page:

1. Create an HTML file that contains a redirect statement to your database layout.

2. Save the HTML file with the .htm or .html extension and place it in the Web folder.

3. In the Web Companion Configuration dialog box, specify the HTML file to be the default home page. (See “Setting Web Companion configuration options” on page 3-3.)

For example, the following redirect statement contains a FileMaker CGI request for layout ID number 3 in the “MyCustomUI.fp5” database to open in the Form View instant web page (formvwcss.htm).

```
<HTML>
<BODY>
        <SCRIPT Language="JavaScript">
                window.location ="/FMRes/FMPJS?-db=MyCustomUI.fp5&
                -layID=3&-token=25&-max=1&-format=formvwcss.htm&
                -mode=browse&-findall"
        </SCRIPT>
</BODY>
```

Note In the Internet Explorer 4.5 for Mac OS browser window, you must allocate at least 6 MB of memory to the web browser in order to display the database layout.

Here’s another example of a redirect statement that displays the database layout in the browser window.

```
<!DOCTYPE HTML PUBLIC “–//W3C//DTD HTML 4.01 Transitional//EN”
        “http://www.w3.org/TR/1999/REC-html1401-19991224/loose.dtd”>
<HTML>
<HEAD>
<META HTTP-EQUIV=”Refresh” Content=”0; URL=/FMRes/FMPJS?
        –db=MyCustomUI.fp5&–layID=3&-token=25&-max=1&
        -format=formvwcss.htm&-mode=browse&-findall”>
</HEAD>
<BODY>
</BODY>
</HTML>
```

Note Layout ID numbers are determined by the original creation order of all the layouts created for the database.

Format filenames for instant web pages

The following table lists the instant web pages and their format filenames as they apply to those web styles that use cascading style sheets.

Instant web page	Format filename
Form View	FormVwCSS.htm
Table View	TableVwCSS.htm
Search	SearchCSS.htm
New Record	NewCSS.htm
Edit Record	EditCSS.htm
Delete Record	DeleteCSS.htm
Sort	SortCSS.htm
Error	Err.htm

Note When you suppress the Instant Web Publishing controls, your users will be completely dependent on your buttons and scripts to manage your database solutions in a browser. You can simulate a “home” link by creating a button to perform a script composed of the single step, Open URL [No dialog, “http:/”]. (Note there is only one “/” after “http:”.) This statement can’t be attached directly to the button itself, but must be included in a script performed by the button.

Web Companion support for Internet media types

The Web Companion supports current MIME (Multipurpose Internet Mail Extensions) types registered for the Internet. The Web Companion can now serve other types of media files besides HTML to the web browser with the correct MIME type.

For example, you might want to serve WML (Wireless Markup Language) documents from your web site. Browsers with the appropriate plug-in would be able to display the file in the browser window.

For information about the Internet media type registry, go to ftp://ftp.iana.org/in-notes/iana/assignments/media-types/.

Monitoring your site

The FileMaker Pro Web Companion generates three types of log files that you can use for gathering information about web users who visit your site:

- access.log
- error.log
- info.log

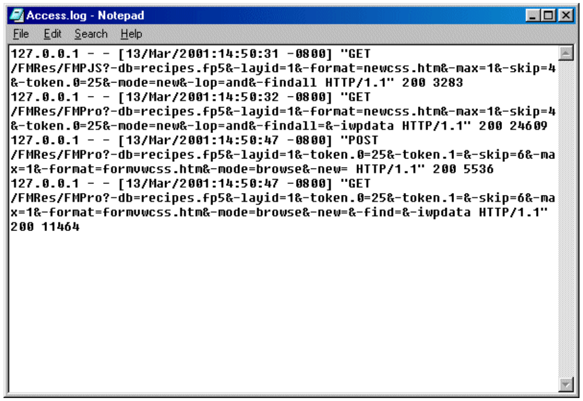
For information on enabling log files, see “Setting Web Companion configuration options” on page 3-3.

In addition, the Web Companion provides several external functions for monitoring activity with your databases, which can be used in your calculation fields and scripts.

Note For information about the log files generated by the Web Server Connector, see “Using the Web Server Connector log files” on page 2-5.

Using the access.log file

The access.log file keeps a record of every time someone accesses the Web Companion from a web browser and lists the hits in NCSA/CERN-compatible Common Log Format.



```
127.0.0.1 - - [13/Mar/2001:14:50:31 -0800] "GET
/FMRes/FMPJS?-db=recipes.fp5&-layid=1&-format=newcss.htm&-max=1&-skip=4
&-token.0=25&-mode=new&-lop=and&-findall HTTP/1.1" 200 3283
127.0.0.1 - - [13/Mar/2001:14:50:32 -0800] "GET
/FMRes/FMPro?-db=recipes.fp5&-layid=1&-format=newcss.htm&-max=1&-skip=4
&-token.0=25&-mode=new&-lop=and&-findall=&-iwpdata HTTP/1.1" 200 24609
127.0.0.1 - - [13/Mar/2001:14:50:47 -0800] "POST
/FMRes/FMPro?-db=recipes.fp5&-layid=1&-token.0=25&-token.1=&-skip=6&-ma
x=1&-format=formvwcss.htm&-mode=browse&-new= HTTP/1.1" 200 5536
127.0.0.1 - - [13/Mar/2001:14:50:47 -0800] "GET
/FMRes/FMPro?-db=recipes.fp5&-layid=1&-token.0=25&-token.1=&-skip=6&-ma
x=1&-format=formvwcss.htm&-mode=browse&-new=&-find=&-iwpdata HTTP/1.1"
200 11464
```

When you enable the Access Log File option in the Web Companion Configuration dialog box, the Web Companion generates an access.log file and places it in the root level of the FileMaker Pro folder.

Every time a web user accesses your database, the Web Companion continuously adds entries to the access.log file.

Note Neither the entries nor the file are automatically deleted, and so the file may become very large. To save hard disk space on your host computer, consider archiving the access.log file on a regular schedule.

The Common Log Format used for the access.log file is:

remotehost rfc931 authuser [date] “request” status bytes

Where	Means this
remotehost	The remote IP address or hostname
rfc931	Required for UNIX systems
authuser	The user name authenticated by the web user
[date]	The date and time of the request

Where	Means this
“request”	The request line exactly as it came from the client
status	The HTTP status code returned to the client (for information, see the World Wide Web Consortium’s web site at www.w3c.org)
bytes	The content length of the document transferred to the client

Using the error.log file

The error.log file, stored in the root level of the folder containing the database, is generated by the Web Companion whenever any unusual errors have occurred. Common errors reported to the web user, such as “Database not open,” are not recorded in the error.log file.

```
[08/Jun/1999:16:16:01:53 –0800] Web Security database not open. Security disabled.

[12/Jul/1999:06:07:02 –0800] ERROR: 6. Could not find email format file.

[23/Jul/1999:11:12:38 –0800] ERROR: 12: Badly formatted URL.
```

Using the info.log file

The info.log file, stored in the root level of the folder for the database, contains entries generated by the [FMP-Log] CDML replacement tag. Whenever web users access FileMaker Pro from your custom CDML web page, information you’ve included within a [FMP-Log] tag is recorded by the Web Companion in the info.log file.

For information about the CDML replacement tags, see chapter 4, “Custom web publishing using CDML.”

Using the Web Companion external functions

You can use the FileMaker Pro Web Companion external functions with your calculations or scripts to:

- check the version of the Web Companion
- capture information about visitors to your database
- translate information in your database to HTML or HTTP

To use a Web Companion external function in a calculation field:

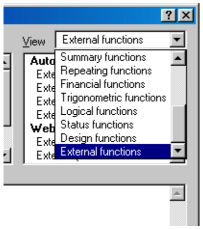
1. Be sure the Web Companion is enabled. (See “Enabling the Web Companion” on page 3-3 for information.)

2. Choose File menu > Define Fields.

3. Type a name for the new calculation field in the Field Name box.

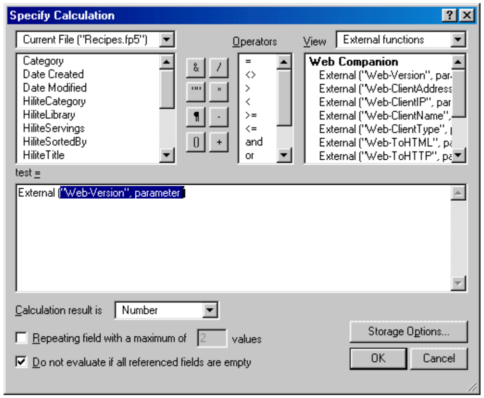
4. For Type, select Calculation, and click Create.

5. In the Specify Calculation dialog box, choose External Functions from the View pop-up menu.



View: External functions

6. Double-click one of the external functions in the list that begins with the function prefix “Web-” to add it to the formula box.



Specify Calculation

Current File ("Recipes.fp5")

View: External functions

test = External ("Web-Version", parameter)

Calculation result is: Number

☐ Repeating field with a maximum of 2 values

☑ Do not evaluate if all referenced fields are empty

Category
Date Created
Date Modified
HiliteCategory
HiliteLibrary
HiliteServings
HiliteSortedBy
HiliteTitle
Operators
&
/
""
×
¶
-
()
+
=
<>
>
<
>=
<=
and
or
Web Companion
External ("Web-Version", par
External ("Web-ClientAddress
External ("Web-ClientIP", par
External ("Web-ClientName",
External ("Web-ClientType", p
External ("Web-ToHTML", pa
External ("Web-ToHTTP", pa
Storage Options...
OK
Cancel

A formula for an external function requires the name of the external function to call and the function’s parameter.

7. Replace the word “parameter” with the required parameter for the function (`0, field name, or text value`).

See the next table for a description of the Web Companion external functions and their parameters.

8. Continue to build the formula as desired and click OK when you’re done.

9. Click Done to close the Define Fields dialog box.

External function’s name and parameter	Description of external function
Web-Version, 0	Returns the version of FileMaker Pro Web Companion that loads when you open FileMaker Pro
Web-ClientAddress, 0	Returns the domain name (for example, www.filemaker.com) of a web user whose HTTP request is currently being processed by the Web Companion. Returns the web user’s IP address if the domain name is not available.
Web-ClientIP, 0	Returns the IP (Internet protocol) address of the web user whose HTTP request is currently being processed by the Web Companion
Web-ClientName, 0	Returns the value that the web user types for a user name in the web browser password dialog box
Web-ClientType, 0	Returns the name and version of the web browser being used by the web user
Web-ToHTML, field name Web-ToHTML, text value	Returns the contents of the specified field or text value encoded in HTML. This is useful, for example, when you want to modify a data field with a calculation and then use it as an error page.
Web-ToHTTP, field name Web-ToHTTP, text value	Returns the contents of the specified field or text value encoded in HTTP. This is useful for fields containing data that should be handled as URLs in HREF links by the browser. For example, “Display Art” becomes “Display%20Art.”

For more information, see chapter 11, “Using formulas and functions,” in the *FileMaker Pro User’s Guide* or see FileMaker Pro Help.

Exporting data to a static HTML page

To quickly display the existing data in your FileMaker Pro database on the Web, you can export it into an HTML table and publish your database as a static web page. To automate the exporting process on a regular basis, use a FileMaker Pro script and control the timing for when the exported table is uploaded to the web site.

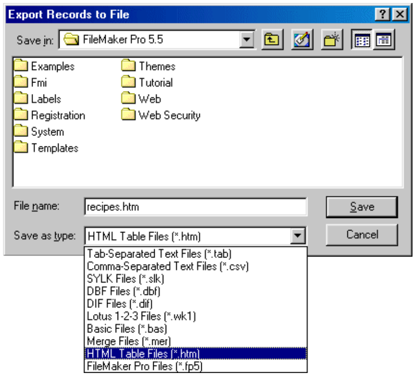
Note Images in container fields cannot be exported. To display them on your web site, place the image files in the Web folder and insert reference links to them in a text field in the database (or insert image links in the exported HTML table).

To export the data into an HTML table:

1. Open the FileMaker Pro database and create a sorted found set with the records you want to export.

If you are going to export subsummary data, include the break field in the sort order.

2. In Browse mode, choose File menu > Export Records. Type a name and select a location for the exported file. Then, choose HTML Table Files (Windows) or HTML Table (Mac OS) for the file type, and click Save.



Export Records to File

Save in: FileMaker Pro 5.5

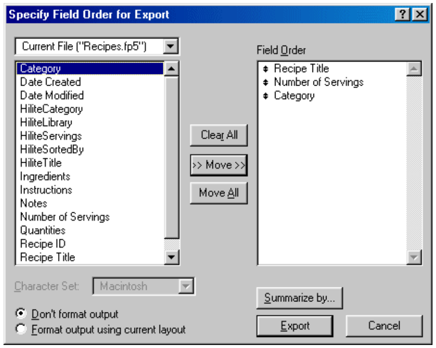
Examples, Fmi, Labels, Registration, System, Templates, Themes, Tutorial, Web, Web Security

File name: recipes.htm

Save as type: HTML Table Files (*.htm)

- Tab-Separated Text Files (*.tab)
- Comma-Separated Text Files (*.csv)
- SYLK Files (*.slk)
- DBF Files (*.dbf)
- DIF Files (*.dif)
- Lotus 1-2-3 Files (*.wk1)
- Basic Files (*.bas)
- Merge Files (*.mer)
- HTML Table Files (*.htm)
- FileMaker Pro Files (*.fp5)

3. In the Specify Field Order for Export dialog box, indicate how you want FileMaker Pro to export the data by selecting fields in the left box and moving them to the Field Order box on the right — in the order that you want them to appear.



Specify Field Order for Export

Current File ("Recipes.fp5")

Category, Date Created, Date Modified, HiliteCategory, HiliteLibrary, HiliteServings, HiliteSortedBy, HiliteTitle, Ingredients, Instructions, Notes, Number of Servings, Quantities, Recipe ID, Recipe Title

Field Order: Recipe Title, Number of Servings, Category

Character Set: Macintosh

● Don't format output
○ Format output using current layout

4. To export a related field, choose the relationship from the pop-up menu.

The related fields for the relationship you chose are then listed in the Specify Field Order for Export dialog box.

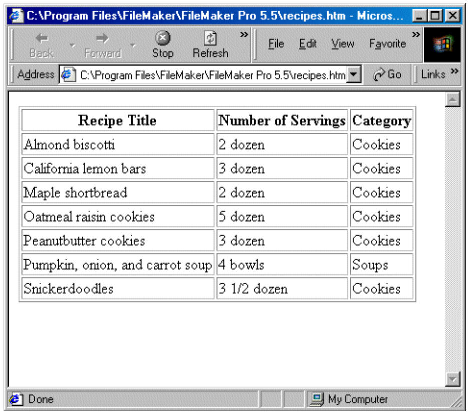
5. To export one record that subtotals multiple records, click Summarize by. Then, in the Summarize by dialog box, click to the left of the field you want to use as the break field. A check mark appears to the left of the field. (You must have set up one or more summary fields and sorted by the break field in the database.) Click OK.

In the Export Field Order for Export dialog box, FileMaker Pro creates italicized temporary export fields in the Field Order list based on the break fields you specified. For example, if you have a summary field named Count and you select the break fields Category and Unit, the export field list contains Count, *Count by Category*, and *Count by Unit*.

6. Select Don't format output if you want to ignore number, date, and time field formatting (for example, to export a number as 3.7 instead of as $3.70).

7. Select Format output using current layout if you want to use number, date, and time field formatting as specified for the included fields on the current layout (for example, to export a number as $3.70 instead of as 3.7). Symbols and other non-numeric values are exported as text.

8. Click Export.



C:\Program Files\FileMaker\FileMaker Pro 5.5\recipes.htm - Micros...

Address: C:\Program Files\FileMaker\FileMaker Pro 5.5\recipes.htm

Recipe Title	Number of Servings	Category
Almond biscotti	2 dozen	Cookies
California lemon bars	3 dozen	Cookies
Maple shortbread	2 dozen	Cookies
Oatmeal raisin cookies	5 dozen	Cookies
Peanutbutter cookies	3 dozen	Cookies
Pumpkin, onion, and carrot soup	4 bowls	Soups
Snickerdoodles	3 1/2 dozen	Cookies

FileMaker Pro saves the data in table format in an HTML file. Each field you specified becomes a column heading in the table and each record becomes a row.

Testing your site without a network connection

You can set up your computer to test the Web Companion and your web site before uploading your site files and databases to the host web server or dialing up to an Internet service provider (ISP).

To set up your computer to act as a single-machine network, enable and activate TCP/IP networking. On Windows machines, this is usually already set up for you (if you can connect to the Internet, then TCP/IP networking is active). On Mac OS machines, one way to set up TCP/IP networking is to create a new TCP/IP control panel configuration that connects via Ethernet to any IP address manually. (For information on setting up TCP/IP networking, see the documentation that came with your operating system.)

Once you've set up your computer to act as a single-machine network, you can type `http://localhost` in your web browser and the FileMaker Pro Web Companion will serve the HTML pages that are located in the Web folder as well as any open databases that are shared via the Web Companion—without connecting to the Internet or intranet.

Tip To thoroughly test your web site, click on every link that exists in your custom web pages under every possible situation, with your databases open, with (and without) any records existing in each database. Did you catch all the errors and create an error message for each of them?

Opening password-protected databases remotely

You can open and close FileMaker Pro databases from your web browser or other client application by making a –dbopen or –dbclose request to FileMaker Pro.

Note You can also open and close FileMaker Pro databases remotely by using the DbOpen and DbClose pseudo procedures with the FileMaker JDBC Driver. See "Using DbOpen and DbClose pseudo procedures" on page 6-5 for information.

The databases must be located in the Web folder and the Web Companion Configuration dialog box must have Remote Administration enabled. In addition, you should require a remote administration password to ensure that once databases are opened, they cannot be closed by an unauthorized user.

The Web Companion uses HTTP basic authentication to enforce web security. When a –dbopen request is made to FileMaker Pro, the browser or client application displays the basic user name/password dialog box where you type `admin` for the user name and the remote administration password that you specified in the Web Companion Configuration dialog box.

For databases that also have an access privileges password, you must use the –password parameter with the –dbopen request. After you enter the admin user name and remote administration password, the Web Companion checks the –password parameter in the request.

Tip For better security, place your databases in subfolders within the Web folder. This way, unauthorized users will not know the rest of the path even if they gain access to the Web folder.

Opening and closing databases using XML

Here is an example of making a –dbopen request using an XML document:

```
FMPro?–db=secretfolder/employees.fp5&–format=–fmp_xml&
–password= dbpassword&–dbopen
```

Here is an example of making a –dbclose request using an XML document:

```
FMPro?–db=secretfolder/employees.fp5&–format=–fmp_xml&–dbclose
```

For more information, see “Generating FileMaker Pro CGI requests for an XML document” on page 5-8 and appendix A, “Valid names used in CGI requests for FileMaker XML data.”

Opening and closing databases using CDML

To open or close databases remotely using a –dbopen request or –dbclose request (CDML variable tags), you must also specify a –format parameter.

If desired, use the [FMP-CurrentError] and [FMP-CurrentDatabase] tags in the format file to display the results of the request (the “current database” was successfully opened or the “current database” could not be opened because of “error #”).

Here is an example of making a –dbopen request:

```
FMPro?–db=secretfolder/employees.fp5&–format=dbopen.htm&
–password=dbpassword&–dbopen
```

Here is an example of making a –dbclose request:

```
FMPro?–db=secretfolder/employees.fp5&–format=dbclose.htm&
–dbclose
```

For more information, see “Generating FileMaker Pro CGI requests using CDML” on page 4-3.

This page intentionally left blank.

Chapter 4
Custom web publishing using CDML

The FileMaker Pro Web Companion supports CDML, a proprietary markup language that enables your HTML pages to interact with a FileMaker Pro database.

The *FMWSC and Tools* CD includes:

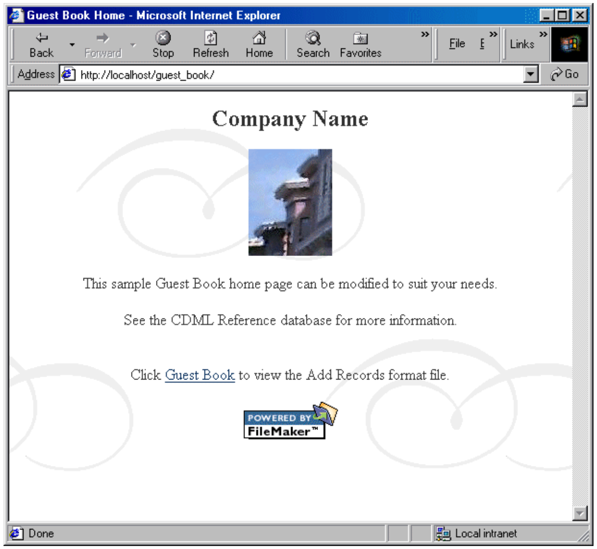
- the CDML Tool database and template files to help you insert the CDML tags into your HTML pages (These HTML pages that contain CDML tags are referred to as *format files*.)
- the CDML Reference database, which describes each CDML tag and how it's used in a format file
- example web sites that publish databases dynamically on the Web using CDML

If you're experienced using CDML, see "Modified CDML tags" on page 4-9 for information about changes to CDML in FileMaker Pro 5.5 and FileMaker Pro 5.5 Unlimited.

Tip You can also publish your FileMaker Pro databases on the Web using open standard XML or via Java applets that use the FileMaker JDBC Driver. See chapter 5, "Using FileMaker Pro XML to deliver your data" and chapter 6, "Using Java and JDBC to deliver your data."

About the CDML examples

FileMaker Pro 5.5 Unlimited provides three examples of databases published on the Web using CDML. The Guest Book example demonstrates how web users can add records to your database by "signing" in a guest book. The Employee Database example is a web site that lets users search the Employees.fp5 database and make modifications to it. The Shopping Cart example lets web users select items from a database and add them to a "shopping cart" for purchase. For information about these CDML examples, see "Planning your web site" on page 4-14.



Address http://localhost/guest_book/

Company Name

This sample Guest Book home page can be modified to suit your needs.

See the CDML Reference database for more information.

Click Guest Book to view the Add Records format file.

Home page of the Guest Book example

For additional information and examples using CDML to publish FileMaker Pro databases on the Web, see the product support pages on the FileMaker, Inc. web site at www.filemaker.com. As a shortcut to the site, double-click FileMaker on the Web on the *FileMaker Pro* CD.

General steps for custom web publishing using CDML

In general, to publish your FileMaker Pro databases on the Web you must have a TCP/IP connection to an intranet or Internet Service Provider, the Web Companion must be enabled, and your databases must be open and shared via the Web Companion. (For more information, see chapter 3, “Publishing your database on the Web.”)

1. Design your web site and how web users will interact with your database. See “Planning your web site” on page 4-14 for ideas.

2. Create a format file for every type of interaction web users will have with your database, for example, adding records to the database. Use the CDML Tool to add tags to your format files and use the format file templates provided by the CDML Tool as a starting point. See “Using the CDML Tool and templates” on page 4-5 for information.

3. Create a custom home page for your web site that contains a link to the first format file you want displayed and the path to the database.

See “Creating a custom home page” on page 3-5 and “Generating FileMaker Pro CGI requests using CDML” on page 4-3 for information.

4. Place the format files and the custom home page in the Web folder, located inside the FileMaker Pro folder on your computer.

5. In FileMaker Pro, enable the Web Companion, open each database that you’re publishing, and share it via the Web Companion.

See “Enabling the Web Companion” on page 3-3 and “Sharing the database via the Web” on page 3-5.

6. In the Web Companion Configuration dialog box, deselect Enable Instant Web Publishing and choose your custom home page from the Home Page pop-up menu if desired. (It must be located in the root level of the Web folder.)

See “Setting Web Companion configuration options” on page 3-3 for information.

7. In a web browser, enter the IP address of the computer that’s hosting the database.

If you enclosed your pages within a site folder, enter the name of the folder after the IP address. Otherwise, the Web Companion will serve the home page that you specified in step 6.

Tip You can create an index page containing a link to every database site that you’re publishing. Each link can contain the IP address and site folder so that users don’t have to type it in their web browsers. When web users click the link, the Web Companion will serve the index.htm file located in the site folder. For example, an index page might have these three links:

<P><A HREF="http://17.17.17.17/guest_book">**Guest Book**</A></P>

<P><A HREF="http://17.17.17.17/employee_database">**Employees**</A></P>

<P><A HREF="http://17.17.17.17/shopping_cart">**Shopping Cart**</A></P>

About CDML format files

A format file is an HTML page containing CDML tags. CDML tags are distinguished by a hyphen (-) or square brackets [ ]. In addition, a format file may contain FileMaker Pro CGI requests in an HTML form or HREF link.

For example, every HTML form in a format file that is requesting data from the database begins with the FMPro form action and the following hidden INPUT tags.

<FORM ACTION=“FMPro” METHOD=“POST”>

```
< INPUT TYPE=“hidden” NAME=“-db” VALUE=“Filename.fp5”>

< INPUT TYPE=“hidden” NAME=“-lay” VALUE=“Layout Name”>

< INPUT TYPE=“hidden” NAME=“-format”
VALUE=“Filename.htm”>

< INPUT TYPE=“hidden” NAME=“-error” VALUE=“Filename.htm”>
```

Note Format files that contain the “FMPro FMRES” form action are instant web pages generated by the Web Companion Instant Web Publishing feature. (See chapter 14, “Publishing databases on the Web,” in the *FileMaker Pro User’s Guide* for information about Instant Web Publishing or see FileMaker Pro Help.)

CDML *variable tags* are used to specify the parameters of a request:

- The names -db and -lay in this example (referred to as CDML variable tags) are used to specify the database and layout for the request.
- The -format name specifies the format file you want the Web Companion to display with the results of the database request.
- The -error name specifies the format file you want displayed in case of an error in the request. (For information on other ways to display an error page, see “Creating error messages” on page 4-13.)

For making requests to the database, the format file must contain a CDML *action tag*. For example, a Delete Record format file contains the -delete action tag in an HTML submit form button.

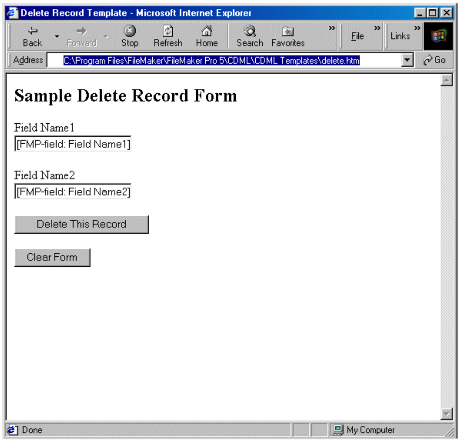
```
< INPUT TYPE=“submit” NAME=“-delete” VALUE=“Delete this
record”>
```

CDML *replacement tags* act as placeholders for data. For example, if the current CDML page is based on a record with “Robert Chan” in FieldName1, the CDML tag [FMP-Field: FieldName1] is replaced with “Robert Chan”.

```
< INPUT TYPE=“text” NAME=“Field Name1” VALUE=“[FMP-Field: Field
Name1]”>
```

Field Name1: Robert Chan

When a format file is displayed statically in the browser rather than as the result of a FileMaker Pro CGI request, CDML replacement tags will appear on the page.



Delete Record Template - Microsoft Internet Explorer

Address C:\Program Files\FileMaker\FileMaker Pro 5\CDML\CDML Templates\delete.htm

Sample Delete Record Form

Field Name1
[FMP-field: Field Name1]

Field Name2
[FMP-field: Field Name2]

The delete.htm template page displayed statically in the browser window

FileMaker Pro 5.5 Unlimited includes templates of the commonly used format files. For information, see “Using the Templates tab” on page 4-6.

Generating FileMaker Pro CGI requests using CDML

You use CDML action tags in FileMaker Pro CGI (Common Gateway Interface) commands to generate requests for data from your database.

For example, to generate a –findall request to display all employees from a database, web users might click on an HREF link containing the following FileMaker Pro CGI command:

```
FMPro?–db=Employees.fp5&–lay=FormView&–format=results.htm
&–findall
```

Or, web users might click on a submit button in an HTML form containing the FMPro form action and the following hidden INPUT elements:

```
<P><FORM ACTION=”FMPro” METHOD=”post”>
<P><INPUT TYPE=”hidden” NAME=”-db” VALUE=”Employees.fp5”>
<P><INPUT TYPE=”hidden” NAME=”-lay” VALUE=”FormView”>
<P><INPUT TYPE=”hidden” NAME=”-format” VALUE=”results.htm”>
```

The submit button in the form contains the –findall request name:

```
<P><INPUT TYPE=“submit” NAME=”-findall” VALUE=”Start Search”>
```

Request names

The name of a request for CDML data is determined by the name of the CDML action tag in the request. You use CDML variable tags to specify the parameters of a request.

Use this request name (CDML action tag)	To generate this request
–delete	Delete record
–duplicate	Duplicate record
–edit	Edit record
–find	Find a record
–findall	Find all records
–findany	find a random record
–new	New record
–dbopen	Open database
–dbclose	Close database
–view	Display format file **Note** The -script variable tag (request parameter) is not designed to work with -view requests.

For a detailed list of the CDML action and variable tags and example syntax for using them in a FileMaker Pro CGI request, see the Tags Index in the CDML Reference database (described in “Using the Tags tab” on page 4-6).

Requests for adding records to a portal

You can use CDML to add records to a portal of related database files. When you make an –edit request or a –new request that includes data for a portal, you must specify the layout and the relationship name for the related database.

Note You can only add one record at a time to a portal, and therefore must make separate –new requests to add more rows to the portal.

The following is an example of a –new request for adding a record to a portal, where “Address::” is the name of the database relationship, and “City.0” is the related field name in the portal:

```
FMPro?–db=Employees.fp5&–lay=LayoutOne&FirstName=Sam&
LastName= Smith&Address::City.0=Seattle&–format=reply.htm&–new
```

Requests for editing multiple records in a portal

You only need to make one –edit request to edit multiple records in a portal. You specify each row (or record) in the portal by adding a period and a consecutive number (starting with number 1) to the end of the related field name.

The following is an example of an –edit request for editing records in a portal, where “Address::” is the name of the relationship, “City.1” is the first row in the portal, and “City.2” is the second row in the portal:

```
FMPro?–db=Employees.fp5&–lay=LayoutOne&recid=11&
FirstName=Sam&LastName=Smith&Address::City.1=Seattle
&Address::City.2=New York&–format=reply.htm&–edit
```

The following is an example of another–edit request for editing records in a portal, in an HTML form:

```
<FORM ACTION="FMPro" METHOD="POST">

    <INPUT TYPE="HIDDEN" NAME="-db"
    VALUE="Employees.fp5">

    <INPUT TYPE="HIDDEN" NAME="-lay" VALUE="LayoutOne">

    <INPUT TYPE="HIDDEN" NAME="-format" VALUE="reply.htm">

    <INPUT TYPE="HIDDEN" NAME="-recid" VALUE="11">

    <INPUT TYPE="TEXT" NAME="FirstName" VALUE="Sam">

    <INPUT TYPE="TEXT" NAME="LastName" VALUE="Smith">

    <INPUT TYPE="TEXT" NAME="Address::City.1"
    VALUE="Seattle">
    <INPUT TYPE="TEXT" NAME="Address::City.2" VALUE="New
    York">
    <INPUT TYPE="SUBMIT" NAME="-edit" VALUE="Edit Record">

</FORM>
```

To display records in a portal on your web page, you use a [FMP-Portal] replacement tag in the format file. For information, see the CDML Reference database (described in “About the CDML Reference database” on page 4-11). For an example of displaying portals in a web page, see the Shopping Cart example (described in “Shopping Cart example” on page 4-17.)

Using the CDML Tool and templates

The CDML Tool is a special FileMaker Pro database that you can use with any open database and your HTML authoring program to copy and paste tags into your format files. It also includes HTML templates for the main types of format files and comments for how to use them.

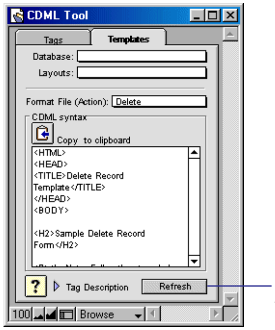
Your databases must be open and shared via the Web Companion. For information, see “Enabling the Web Companion” on page 3-3 and “Sharing the database via the Web” on page 3-5.

When you specify the name of an open database in the CDML Tool, the database’s layout, field, and value list names will automatically appear in pop-up menus for you to choose from. Other database-specific information will appear as appropriate within the CDML tags.

To use the CDML Tool:

1. Open the CDML_Tool.fp5 database located in the CDML folder.

FMWSC and Tools > CDML > Web Tools > CDML Tool.fp5



Database:

Layouts:

Format File (Action): Delete

CDML syntax

```
<HTML>
<HEAD>
<TITLE>Delete Record
Template</TITLE>
</HEAD>
<BODY>

<H2>Sample Delete Record
Form</H2>
```

Click to update list options for the selected database

The Templates tab in the CDML Tool

2. Open the database that you’re planning to publish on the Web.

3. In a text editor or HTML authoring program, create a blank page.

4. Arrange the windows so you can easily go between the blank page and the CDML Tool.

5. In the CDML Tool, select either the Templates tab or the Tags tab and click the Refresh button. Then choose your database from the Database pop-up menu, and the layout that you want to use from the Layouts pop-up menu.

The layout you choose determines which fields will be accessed. Otherwise, it does not affect the layout of the web page.

Using the Templates tab

1. With the Templates tab selected, choose a format file from the Format File (Action) pop-up menu for the type of action that you want users to perform on your database.

For a description of each format file, see “Customizing a format file template” on page 4-7.

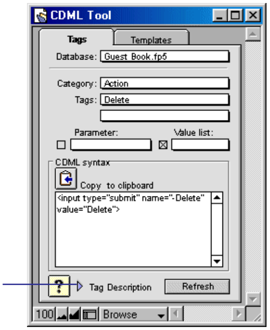
2. Click the Copy to clipboard button.

3. Paste the template into your blank HTML page. (If you’re using an HTML authoring program, make sure to paste the template into the HTML source.)

Using the Tags tab

1. With the Tags tab selected, choose a category from the Category pop-up menu and then click the Refresh button to update the type of tags listed in the Tags pop-up menu.

For a description of the categories, see “Categories of CDML tags” on page 4-8.



Database: Guest Book.fp5

Category: Action

Tags: Delete

□ ☒

```
<input type="submit" name="-Delete"
value="Delete">
```

Click to expand the Tag Description box

The Tags tab in the CDML Tool

2. Choose a tag from the Tags pop-up menu.

If the tag has a corresponding list of parameters or value list options, those values will appear in the Parameters and Value List pop-up menus.

3. Choose a parameter or value list option, if applicable, and click the check box on the left to select your choice. When a value list option is selected, the parameter appears in the CDML Syntax box.

For example, the Check boxes tag in the Fields (Add) Dynamic category includes a list of encoding parameters you can choose from (Raw, HTML, Break, Display, and URL) and a choice of two value list parameters (Value List Name or Yes/No).

[FMP-ValueList: Value List Name]
[FMP-ValueList: Yes/No]

The HTML parameter is selected by default for the [FMP-Field] replacement tag.

For information, see “Using an encoding parameter with a CDML replacement tag” on page 4-14 and the individual tag descriptions in the Tag Index tab of the CDML Reference (described in “About the CDML Reference database” on page 4-11).

4. View the tag’s syntax in the box at the bottom of the CDML Tool window.

You can also click the Tag Description arrow to expand the window and view a description of the tag in another box below.

5. Click the Copy to clipboard button.

6. Paste the tag into your format file.

Customizing a format file template

The CDML Tool provides nine format file templates for the basic type of actions that web users can perform with your database. These templates are also included in the CDML Templates folder on the *FMWSC and Tools* CD.

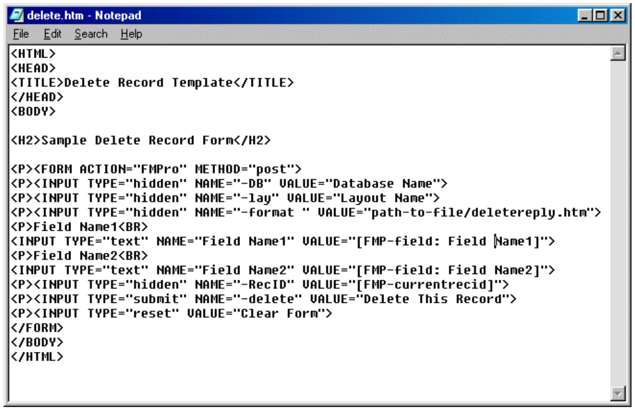
Choose this template	So web users can do this with your database
Delete.htm	Delete a selected record
Delete_reply.htm	Receive feedback after deleting a record
Detail.htm	View one record in detail on the screen
Edit.htm	Make changes to one record
Edit_reply.htm	Receive feedback after editing a record
New.htm	Add a record
New_reply.htm	Receive feedback after adding a record
Results.htm	View the results of a search for records
Search.htm	Search for specific records

To use a format file template:

1. Copy and paste the template from the CDML Tool into the HTML source of a blank document, or open the template text file from the CDML Templates folder (inside the CDML folder) in a text editor or HTML authoring program.

FMWSC and Tools > CDML > Web Tools > CDML Templates

Note When you're familiar with the template, you may want to delete the comments (including the comment tags <!-- -->).



```
<HTML>
<HEAD>
<TITLE>Delete Record Template</TITLE>
</HEAD>
<BODY>

<H2>Sample Delete Record Form</H2>

<P><FORM ACTION="FMPro" METHOD="post">
<P><INPUT TYPE="hidden" NAME="-DB" VALUE="Database Name">
<P><INPUT TYPE="hidden" NAME="-lay" VALUE="Layout Name">
<P><INPUT TYPE="hidden" NAME="-format " VALUE="path-to-file/deletereply.htm">
<P>Field Name1<BR>
<INPUT TYPE="text" NAME="Field Name1" VALUE="[FMP-field: Field Name1]">
<P>Field Name2<BR>
<INPUT TYPE="text" NAME="Field Name2" VALUE="[FMP-field: Field Name2]">
<P><INPUT TYPE="hidden" NAME="-RecID" VALUE="[FMP-currentrecid]">
<P><INPUT TYPE="submit" NAME="-delete" VALUE="Delete This Record">
<P><INPUT TYPE="reset" VALUE="Clear Form">
</FORM>
</BODY>
</HTML>
```

The delete.htm format file with all <!--comments--> removed

2. Use the CDML Tool to add any CDML tags for the fields and other types of data you're requesting from the database.

See "Categories of CDML tags" next.

3. Add text, images and links as desired.

4. Name and save the file in your web site.

This template	Contains these CDML tags
Delete.htm	-db, -lay, -format, [FMP-Field: Field Name1], [FMP-Field: Field Name2], [FMP-CurrentRecID], and -delete
Delete_reply.htm	[FMP-LinkRecID: Layout=Layout Name, Format=path-to-file/default.htm]
Detail.htm	-db, -recid, [FMP-CurrentRecID], -lay, -format, [FMP-Field: Field Name1], [FMP-Field: Field Name2], -edit
Edit.htm	-db, -recid, [FMP-CurrentRecID], -lay, -format, [FMP-Field: Field Name1], [FMP-Field: Field Name2], -edit
Edit_reply.htm	[FMP-LinkRecID: Layout=Layout Name, Format=path-to-file/Detail.htm]
New.htm	-db, -lay, -format, -new

This template	Contains these CDML tags
New_reply.htm	[FMP-LinkRecID: Layout=Layout Name, Format=path-to-file/Detail.htm]
Results.htm	[FMP-RangeStart], [FMP-RangeEnd], [FMP-CurrentFoundCount], [FMP-RangeSize], [FMP-LinkPrevious], [/FMP-LinkPrevious] [FMP-Record] [FMP-Field: Field Name1], [FMP-Field: Field Name2] [FMP-LinkRecID: Layout=layout_name, Format=path-to-file/detail.htm] [/FMP-Record] [FMP-LinkNext], [/FMP-LinkNext]
Search.htm	-db, -lay, -format, -op -find

For examples of more complex format files, see the CDML examples included with FileMaker Pro 5.5 Unlimited. These examples are described in “Planning your web site” on page 4-14.

Categories of CDML tags

There are three types of FileMaker CDML tags:

- Action tags — these tags are used to make a specific request to the database, such as to add a record. Action tags always begin with a hyphen, such as the -new tag. (See “Generating FileMaker Pro CGI requests using CDML” on page 4-3 for information.)
- Variable tags — these tags are used to specify the parameters of a request, such as the names of the database and the layout. They also begin with a hyphen, such as the -db and -lay tags.
- Replacement tags — these tags are used to display data from the database on a web page. They act as placeholders until a request has been submitted and the requested data is returned to the page. Replacement tags always begin and end with a square bracket, for example, [FMP-Field: First Name].

The CDML Tool organizes the CDML tags and HTML form tags into 15 categories:

Use tags in this category	For this type of interaction with FileMaker Pro
Action	Delete, duplicate, edit, find, or add a record, find all records, find a random record, reset a form, or view a format file containing replacement tags or value lists
Email	Display an email form (containing BCC, CC, Format, From, Host, Subject, To, or All Mail Tags)
Fields (Add) Dynamic	Display dynamic field name and value list information in HTML form elements (text field, text area, check box, radio button, repeating fields, scrolling list) and CDML replacement tags such as [FMP-ValueList] so web users can view and modify current FileMaker Pro data on the web page. When changes are made in the database, the current information is displayed on the web page.
Fields (Add) Static	Display static field name and value list information from the database in HTML form elements (text field, text area, check box, radio button, repeating fields, scrolling list). This information does not change on the web page regardless of changes made in the database.
Fields (Display)	Display dynamic field name and value list information in HTML form elements (text field, text area, check box, radio button, repeating fields, scrolling list) and display field data in [FMP-Field], [FMP-ValueListItem], and [FMP-RepeatingItem] replacement tags. Web users cannot edit this data.
Fields (Update)	Display dynamic field name and value list information in HTML form elements (text field, text area, check box, radio button, repeating fields, scrolling list) and display field data in [FMP-Field], [FMP-ValueListItem], [FMP-RepeatingItem], and [FMP-Option] replacement tags within INPUT and SELECT elements so web users can edit this data
Find Operators	Use an operator (AND/OR, OR, hidden, number/dates, or text) when performing a –find request

Use tags in this category	For this type of interaction with FileMaker Pro
Hidden	Do not display this INPUT tag (–db, –lay, –db & –lay, –error, –format, or –token)
Links	Display a format file based on CDML replacement or action tags within HREF, MAILTO, and SRC (image) links
Logical	Conditionally display data within the [FMP-If] and [FMP-Else] replacement tags
Looping	Display multiple lines of data within one of these looping types of replacement tags (Current Find, Current Sort, Layout Fields, Portal, Record, Repeating Fields, Value List, or Value Names)
Names Only	Display a list of database, field, layout, script, or value list names from any open database
Replacement	Display specific data from the database in one of 44 types of replacement tags on the web page. For example, display the web user's IP address in the [FMP-ClientIP] replacement tag.
Variables (Add)	Generate information from the client (web user's) computer based on one of eight replacement tags received as parameters to a request in the FileMaker Pro CGI command: [FMP-ClientAddress], [FMP-ClientIP], [FMP-ClientType], [FMP-ClientUserName], [FMP-CurrentDate], [FMP-CurrentTime],[FMP-CurrentDay], and [FMP-CurrentToken]
Variables (Display)	Display information in one of 21 replacement tags that correspond to a specific request parameter. For example, display the maximum number of records in the [FMP-MaxRecords] replacement tag as specified by the –max request parameter (CDML variable tag).

Modified CDML tags

The FileMaker Pro 5 Web Companion supported the following new or modified CDML tags. Example syntax for these tags is described in the CDML Reference database (see "About the CDML Reference database" on page 4-11 for information).

Note The –fmtfield, –mailfmtfield, and –errorfmtfield variable tags have been disabled and are no longer available for use because of the security risk they posed for databases published on the Web.

CDML tags new to FileMaker Pro 5

New CDML action tags	Description
–dbopen (action)	Open a database that's located in the Web folder. Required parameters: -db, and -format variable tags Optional parameter: -password (Remote administration privileges must be set in the Web Companion Configuration dialog box.) See "Opening password-protected databases remotely" on page 3-18 for more information.
–dbclose (action)	Close a database that's open in the Web folder or in a site folder within the Web folder. Required parameters: -db, and -format variable tags (Remote administration privileges must be set in the Web Companion Configuration dialog box.) See "Opening password-protected databases remotely" on page 3-18 for more information.

New CDML variable tags	Description
–errnum (variable)	Specify a range or single error number to be handled by the -error variable tag. You can specify discontinuous ranges, such as -errnum=500-510& -errnum=900-978. If no error numbers are specified, all errors are handled by the –error tag. See appendix B "FileMaker Pro values for error codes" for a list of error numbers.
–modid (variable)	Detect record modification collisions by including the –modid variable with the –edit action tag. If the modification ID does not match in an –edit request, then a new error code 306 is returned (see appendix B, "FileMaker Pro values for error codes").

New CDML replacement tags	Description
[FMP-CurrentModID] (replacement)	Used in conjunction with the –modid request parameter to replace with the record's current internal modification ID. The [FMP-If] tag supports the CurrentModID and CurrentRecID parameters as numeric types.
[FMP-CurrentPortalRowNumber] (replacement)	Replace with the number of the current portal row. Use this tag within the [FMP-Portal] looping tag.
[FMP-CurrentRepeatNumber] (replacement)	Replace with the number of the current repetition in a repeating field. Use this tag within the [FMP-Repeating] looping tag.
[FMP-ElseIf] (replacement)	Replace with specified data if the conditions of the [FMP-If] are not met. Use this tag with the [FMP-If] tag and a conditional operator.

New CDML replacement tags	Description
[FMP-InlineAction] (replacement)	Lets you send multiple requests to your database from within a single format file. The –format request parameter (CDML variable tag) is ignored within the inline request.
[FMP-Log] (replacement)	Lets you specify comments to be recorded in the Info.log file each time the page that contains this tag is accessed through the Web Companion . (See "Web Companion support for Internet media types" on page 3-13 for information.)

Modified CDML tags

Modified CDML tag	Description
-token (variable)	You can use multiple tokens by specifying a number from 0 to 9 after a period in the variable, for example, -token.3.
[FMP-CurrentToken] (replacement)	This replacement tag will now accept a parameter value of 0 to 9, for example, [FMP-CurrentToken: 3].
[FMP-If: CurrentToken] (modified parameter)	The currenttoken parameter of the [FMP-If] tag can now include a number from 0 to 9, for example [FMP-If: CurrentToken:3.eq.{CurrentToken:4}].
[FMP-If: CurrentPortalRowNumber] (new parameter)	Use this parameter with the [FMP-If] tag to specify a number for the current portal row within the [FMP-Portal] looping tag.
[FMP-If: CurrentRepeatNumber] (new parameter)	Use this parameter with the [FMP-If] tag to specify a number for the current field repetition within the [FMP-Repeating] looping tag.

Modified CDML tag	Description
[FMP-If:.and.] [FMP-If:.or.] [FMP-If:.xor.] (new parameters)	Use these parameters (conditional operators) with the [FMP-If] tag to specify multiple replacement conditions. If desired, use this modified [FMP-If] tag along with the new [FMP-ElseIf] replacement tag.
[FMP-Field: FieldName, Format] and [FMP-RepeatingItem: FieldName, Format] (new encoding parameter)	Use the Format parameter to generate a higher resolution HTML for field data. Numbers, dates, times, and container size are formatted according to the layout's format in the database. If no size is specified in the layout for container fields, then the size is based on the size of the objects (pictures or QuickTime movies) within the fields. The Format parameter should not be used with fields inside TEXT INPUT or TEXT AREA elements in an HTML form. The Format parameter supports Bold, Italic, Underline, Strikethrough, Superscript, Subscript, Uppercase, and Lowercase styles in most browsers.
	FileMaker Pro font sizes map to these HTML font sizes: 8 points or less = -3 9 points = -2 10 points = -1 11 - 13 points = 0 (SIZE is not used) 14 - 17 points = +1 18 - 23 points = +2 24 points or greater = +3

Using an intratag parameter

You can add a parameter to certain replacement tags that is based on the contents of fields and other items.

Any first parameter that is allowed with the [FMP-If] tag on the left side of an operator, such as CanDelete, ClientPassword or ValueListItem, can be used as a third parameter on the right side of an operator with these replacement tags, as long as they're within curly brackets { }. (See the [FMP-If] tag syntax in the CDML Reference for a description of the first parameters for the [FMP-If] tag.)

Note Some restrictions apply to this new intratag parameter. See the CDML Reference for a description of each replacement tag and how the intratag parameter may be used.

Modified replacement tags that allow for the intratag parameter include:

[FMP-If: {intratag}]

[FMP-Cookie: {intratag}]

[FMP-InlineAction: {intratag}]

[FMP-Log: {intratag}]

[FMP-SetCookie: {intratag}]

[FMP-ContentMimeType: {intratag}]

[FMP-CurrentToken: {intratag}]

[FMP-LinkRecID: {intratag}]

[FMP-Field: {intratag}]

[FMP-Option: {intratag}]

[FMP-Repeating: {intratag}]

[FMP-ValueList: {intratag}]

About the CDML Reference database

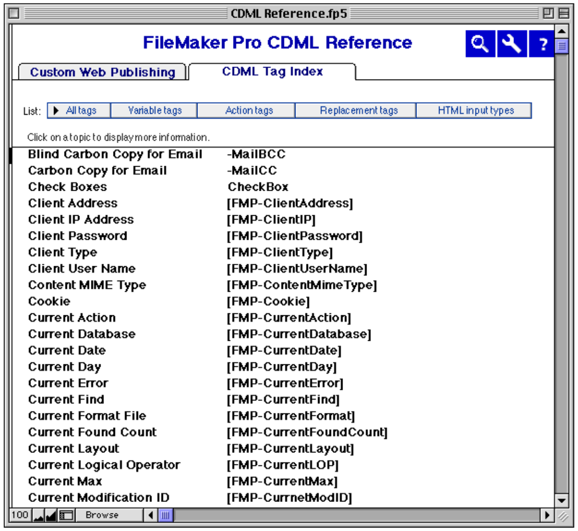
The CDML Reference database is divided into two parts:

- CDML Tag Index — an index for all of the CDML tags with topics that describe what each tag does and how it appears in syntax
- Custom Web Publishing — general guidelines for custom web publishing using CDML

To view the CDML Reference database:

1. In FileMaker Pro, open the CDML_Reference.fp5 file located inside the CDML folder.

FMWSC and Tools > CDML > Web Tools > CDML Reference.fp5



CDML Reference.fp5

FileMaker Pro CDML Reference

Custom Web Publishing | CDML Tag Index

List: All tags | Variable tags | Action tags | Replacement tags | HTML input types

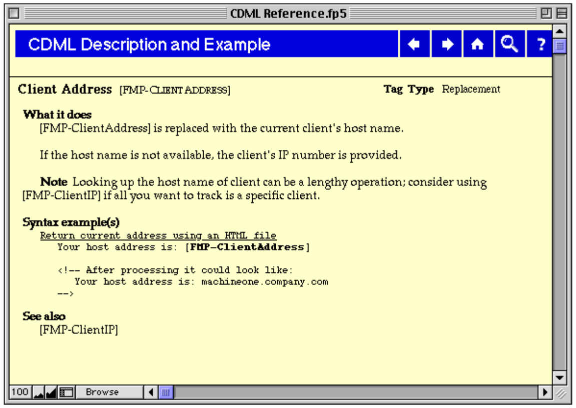
Click on a topic to display more information.

Blind Carbon Copy for Email	-MailBCC
Carbon Copy for Email	-MailCC
Check Boxes	CheckBox
Client Address	[FMP-ClientAddress]
Client IP Address	[FMP-ClientIP]
Client Password	[FMP-ClientPassword]
Client Type	[FMP-ClientType]
Client User Name	[FMP-ClientUserName]
Content MIME Type	[FMP-ContentMimeType]
Cookie	[FMP-Cookie]
Current Action	[FMP-CurrentAction]
Current Database	[FMP-CurrentDatabase]
Current Date	[FMP-CurrentDate]
Current Day	[FMP-CurrentDay]
Current Error	[FMP-CurrentError]
Current Find	[FMP-CurrentFind]
Current Format File	[FMP-CurrentFormat]
Current Found Count	[FMP-CurrentFoundCount]
Current Layout	[FMP-CurrentLayout]
Current Logical Operator	[FMP-CurrentLOP]
Current Max	[FMP-CurrentMax]
Current Modification ID	[FMP-CurrnetModID]

2. On the CDML Tag Index tab, click a button to see a list of tags grouped by one of these categories: All tags, Action tags, Variable tags, Replacement tags, or HTML Input Types (form elements).

3. Then click a tag to display a topic about it.

Each topic explains how the CDML tag is used and its syntax.



CDML Reference.fp5

CDML Description and Example

Client Address [FMP-CLIENT ADDRESS] **Tag Type** Replacement

What it does

[FMP-ClientAddress] is replaced with the current client's host name.

If the host name is not available, the client's IP number is provided.

Note Looking up the host name of client can be a lengthy operation; consider using [FMP-ClientIP] if all you want to track is a specific client.

Syntax example(s)

```
Return current address using an HTML file
   Your host address is: [FMP-ClientAddress]

   <!-- After processing it could look like:
      Your host address is: machineone.company.com
   -->
```

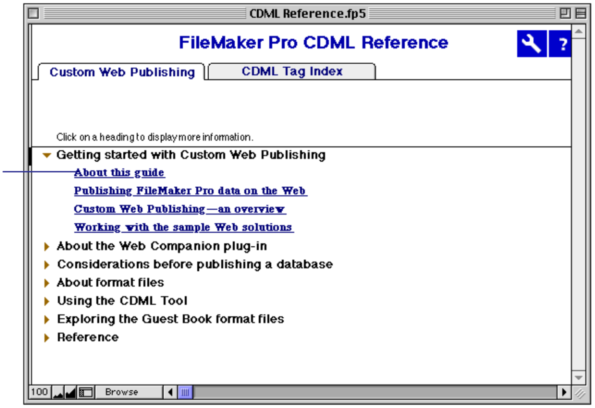
See also

[FMP-ClientIP]

4. Click the home button to return to the FileMaker Pro CDML Reference screen.

From the home screen, you can open the CDML Tool by clicking the tool button.

5. Click the Custom Web Publishing tab to view general information about custom web publishing using CDML.



CDML Reference.fp5

FileMaker Pro CDML Reference

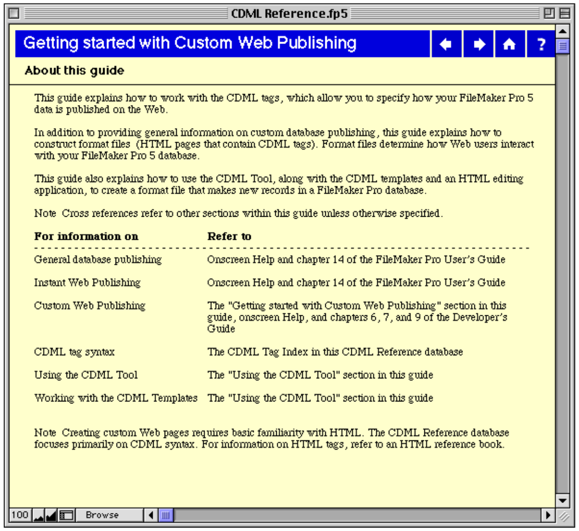
Custom Web Publishing | CDML Tag Index

Click on a heading to display more information.

- Getting started with Custom Web Publishing
  - About this guide
  - Publishing FileMaker Pro data on the Web
  - Custom Web Publishing—an overview
  - Working with the sample Web solutions
- About the Web Companion plug-in
- Considerations before publishing a database
- About format files
- Using the CDML Tool
- Exploring the Guest Book format files
- Reference

Click to view a topic

6. Click an arrow or a subject title in the table of contents to view a list of underlined topic titles. Then click an underlined title to view the topic.



Getting started with Custom Web Publishing

About this guide

This guide explains how to work with the CDML tags, which allow you to specify how your FileMaker Pro 5 data is published on the Web.

In addition to providing general information on custom database publishing, this guide explains how to construct format files (HTML pages that contain CDML tags). Format files determine how Web users interact with your FileMaker Pro 5 database.

This guide also explains how to use the CDML Tool, along with the CDML templates and an HTML editing application, to create a format file that makes new records in a FileMaker Pro database.

Note Cross references refer to other sections within this guide unless otherwise specified.

For information on	**Refer to**
General database publishing	Onscreen Help and chapter 14 of the FileMaker Pro User's Guide
Instant Web Publishing	Onscreen Help and chapter 14 of the FileMaker Pro User's Guide
Custom Web Publishing	The "Getting started with Custom Web Publishing" section in this guide, onscreen Help, and chapters 6, 7, and 9 of the Developer's Guide
CDML tag syntax	The CDML Tag Index in this CDML Reference database
Using the CDML Tool	The "Using the CDML Tool" section in this guide
Working with the CDML Templates	The "Using the CDML Tool" section in this guide

Note Creating custom Web pages requires basic familiarity with HTML. The CDML Reference database focuses primarily on CDML syntax. For information on HTML tags, refer to an HTML reference book.

7. Click an arrow button to continue reading the topics in sequence.

Creating error messages

The FileMaker Pro Web Companion automatically generates an error page for various errors that occur when web users access a database. You can specify your own error pages to override the built-in pages in the following ways:

- Use the –error variable tag in a FileMaker Pro CGI request to specify a single error page, which will be displayed for every type of error that occurs.
- Use the –error variable tag in the request to specify an error page that contains [FMP-If] and [FMP-Else] replacement tags and FileMaker Pro error codes, which determine certain conditions for displaying an error message.
- Include –errnum along with the –error tag in the request to specify a range or single error code. You can use multiple –errnum tags to specify discontinuous ranges, such as 500-510 and 900-978.
- Include a specific error page with a filename that the Web Companion recognizes when it's located in a folder named "Error" inside the Web folder.

The –error tag in a FileMaker Pro CGI request overrides both the built-in error pages and the pages within the Error folder.

For any error page not included in the Error folder, the Web Companion generates a built-in error page instead.

The Web Companion recognizes these error pages in the Error folder:

NoResults.htm

ReqFieldMissing.htm

RepRelatedField.htm

UnexpectedError.htm

NotEnoughMemory.htm

DatabaseNotOpen.htm

LayoutNotFound.htm

DataEntryError.htm

DatabaseViolation.htm

FieldViolation.htm

SecurityDisabled.htm

FormatNotFound.htm

FileNotFound.htm

InvalidAddress.htm

For a complete list of error code numbers, see appendix B, “FileMaker Pro values for error codes.”

For examples of error pages, see the gb_error.htm file in the Guest Book example, the errors.htm file in the Employee Database example, and the reqfielderror.htm file in the Shopping Cart example. (See “Looking at the three CDML examples” on page 4-15 for more information.)

FMWSC and Tools > CDML > CDML Examples

Using an encoding parameter with a CDML replacement tag

You can use a special encoding parameter with certain CDML replacement tags to specify how the data will be encoded by the web browser when it is sent to the web page.

```
[FMP-Field: Information, Break]
```

This encoding parameter	Tells the browser to do this
Raw	Don’t perform any encoding
HTML	Use standard HTML encoding
Format (new parameter)	Use standard HTML encoding and replace text, numbers, dates, times, and container size formatting with FileMaker Pro formats (for example, to add <BOLD> and <ITALIC> elements) This new parameter can be used with the [FMP-Field] and [FMP-RepeatingItem] replacement tags. (See “Modified CDML tags” on page 4-9.)
Break	Use standard HTML encoding and replace soft carriage returns with the <BR> HTML element
Display	Translate text to the language specified in the Web Companion Configuration dialog box. Use this parameter with the [FMP-CurrentAction], [FMP-FindOpItem], and [FMP-SortOrderItem] replacement tags. For information, see the CDML Reference database (described in “About the CDML Reference database” on page 4-11) and “Setting Web Companion configuration options” on page 3-3.
URL	Use URL encoding, which includes converting spaces to %20.

The encoding parameters are displayed in the Parameters pop-up menu in the CDML Tool when the appropriate CDML replacement is selected. (See “Using the Tags tab” on page 4-6.)

Planning your web site

You’ll need to create a format file for every type of interaction with FileMaker Pro that you want your web site to provide. As you create each format file, you’ll need to know

- what web users will do on the page
- what type of requests will be made to the database
- which format file will be displayed as a result of each request

You’ll also need to create pages to display error messages and other types of feedback.

As you add CDML tags to a format file, you’ll need to know in advance what the names are of the database, the layout you want to use, and the next format file in the sequence. A flowchart can be useful to map out the page links and interactions with each database.

Three examples are included with FileMaker Pro 5.5 Unlimited to give you ideas on how to organize a site. You can also use them as templates and modify them for your own sites.

Looking at the three CDML examples

The *FMWSC and Tools* CD includes three CDML examples that demonstrate ways to publish your databases on the Web using CDML. These examples are located in the CDML folder:

FMWSC and Tools > CDML > CDML Examples

To examine the three CDML examples:

1. Copy the CDML Examples folder and its contents into the root level of the Web folder, located inside the FileMaker Pro 5.5 application folder. The contents include one default.htm file, an Images folder, and the Employee Database, Guest Book, and Shopping Cart folders.

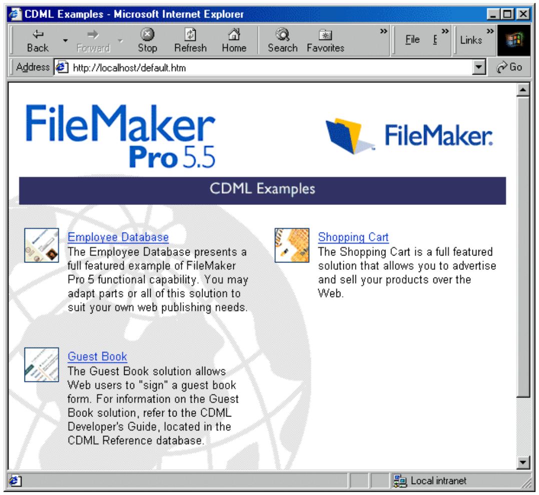
2. In FileMaker Pro, open each database file inside the three example folders and make sure that each one is shared via the Web Companion. See “Enabling the Web Companion” on page 3-3 and “Sharing the database via the Web” on page 3-5 for information.)

3. In your web browser, type `http://localhost` or your computer’s IP address followed by `/CDML Examples/` and press Enter.

http://localhost/CDML Examples/

The Web Companion displays the default.htm file located inside the CDML Examples folder.

For information on setting up your computer as a localhost, see “Testing your site without a network connection” on page 3-18.



Employee Database
The Employee Database presents a full featured example of FileMaker Pro 5 functional capability. You may adapt parts or all of this solution to suit your own web publishing needs.

Shopping Cart
The Shopping Cart is a full featured solution that allows you to advertise and sell your products over the Web.

Guest Book
The Guest Book solution allows Web users to "sign" a guest book form. For information on the Guest Book solution, refer to the CDML Developer's Guide, located in the CDML Reference database.

Click a link to go to the default.htm file for each CDML example

4. Click the links on the Web Companion Demonstration page to go to each of the three examples.

5. As you explore each example site, view the HTML source on each page to see how CDML is used.

Be sure to examine the FileMaker Pro CGI commands in the HREF links or HTML forms. (See “Generating FileMaker Pro CGI requests using CDML” on page 4-3.)

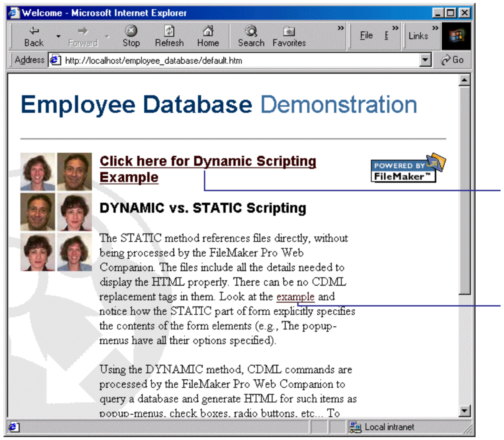
Note To see the CDML replacement tags, open the example files directly in the browser (without using the web site example links).

Employee Database example

The Employee Database example is designed to demonstrate the most widely used CDML tags. It shows how to

- search for records in the database
- select sorting options

- browse the database
- preselect how many records to view at a time
- add a record to the database



Employee Database Demonstration

Click here for Dynamic Scripting Example

DYNAMIC vs. STATIC Scripting

The STATIC method references files directly, without being processed by the FileMaker Pro Web Companion. The files include all the details needed to display the HTML properly. There can be no CDML replacement tags in them. Look at the example and notice how the STATIC part of form explicitly specifies the contents of the form elements (e.g., The popup-menus have all their options specified).

Using the DYNAMIC method, CDML commands are processed by the FileMaker Pro Web Companion to query a database and generate HTML for such items as popup-menus, check boxes, radio buttons, etc... To

Click to search, browse, and add records to the Employees.fp5 database

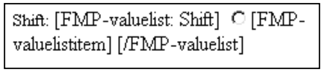
Click to display the example.htm page statically

The Employee Database example also provides a demonstration of the difference between using a static HTML form and dynamic CDML to display a value list. Examine the difference between two example links that both go to the same example.htm page but only one dynamically displays value lists (data) from the database.

The first example link is a simple HREF link to the example.htm page.

<A HREF="example.htm">example</A>

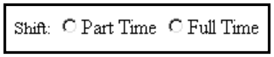
And the CDML replacement tags appear statically on the page.


Shift: [FMP-valuelist: Shift] ○ [FMP-valuelistitem] [/FMP-valuelist]

The second example link contains a FileMaker Pro CGI command for a CDML -view request. (The -view CDML action tag is usually used to display format files containing value lists or CDML replacement tags.)

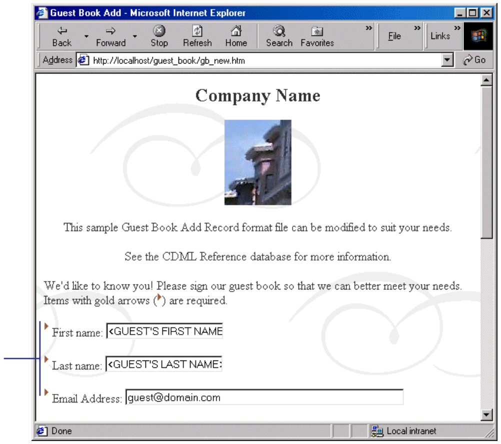
<A HREF=FMPro?-db=Employees.fp5&-lay=Detail&-format=example.htm&-view">example</A>

The CDML replacement tags are replaced with the value lists from the database.


Shift: ○ Part Time ○ Full Time

Guest Book example

The Guest Book example demonstrates how web users can enter information on a web page (in an HTML form) and submit it to a new record in a FileMaker Pro database. The HTML form that is used on the New Record web page includes CDML tags that allow the database to be accessed and a new record to be created.


Guest Book Add - Microsoft Internet Explorer
Back
Forward
Stop
Refresh
Home
Search
Favorites
File
Links
Address
http://localhost/guest_book/gb_new.htm
Go
Done
Local intranet

Company Name

This sample Guest Book Add Record format file can be modified to suit your needs.

See the CDML Reference database for more information.

We'd like to know you! Please sign our guest book so that we can better meet your needs. Items with gold arrows (‣) are required.

First name: <GUEST'S FIRST NAME

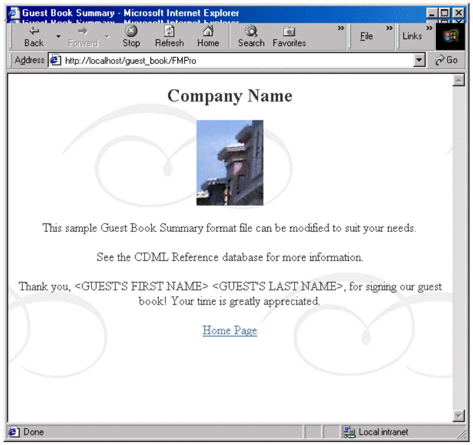
Last name: <GUEST'S LAST NAME:

Email Address: guest@domain.com

Arrows indicate required fields in the database

The New Record page in the Guest Book example

Once a guest signs in and sends the information, a summary page is returned notifying the guest that the information has been received. The summary page contains [FMP-Field] replacement tags for the first and last name of the guest that was entered in the new record.


Guest Book Summary - Microsoft Internet Explorer
Back
Forward
Stop
Refresh
Home
Search
Favorites
File
Links
Address
http://localhost/guest_book/FMPro
Go
Done
Local intranet

Company Name

This sample Guest Book Summary format file can be modified to suit your needs.

See the CDML Reference database for more information.

Thank you, <GUEST'S FIRST NAME> <GUEST'S LAST NAME>, for signing our guest book! Your time is greatly appreciated.

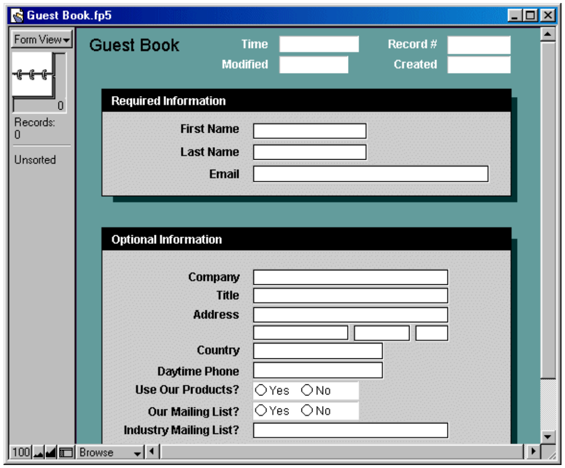
Home Page

The Guest Book summary page shows the web user’s first and last names from the new record

Arrow icons on the New Record page indicate three fields that are required fields in the database. If any of these fields is left blank when the information is submitted to the database, then the error message page (gb_err.htm) is displayed. The error page is specified on the New Record page in a hidden INPUT element.

```
<INPUT TYPE="hidden" NAME="-error" VALUE="gb_err.htm">
```

For information, see “Creating error messages” on page 4-13.



Guest Book

Time
Modified
Record #
Created

Required Information

First Name
Last Name
Email

Optional Information

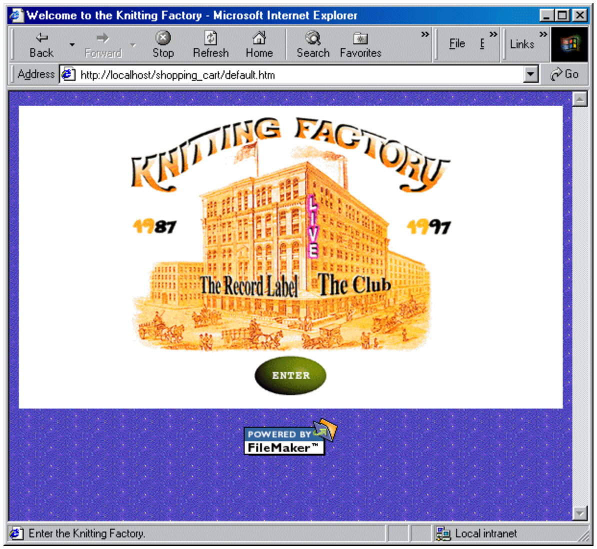
Company
Title
Address
Country
Daytime Phone
Use Our Products? ○ Yes ○ No
Our Mailing List? ○ Yes ○ No
Industry Mailing List?

Web users add records to the empty Guest_Book.fp5 database

Shopping Cart example

The Shopping Cart example is a complex web site for the Knitting Factory record label company. The site includes five databases for searching and browsing the company’s product list, listening to audio demos, and placing orders.

The site keeps track of each web user by passing CDML tokens from one web page to the next. Some of the CDML format files use the [FMP-Include] replacement tag to reference text files and include their content on the web pages. In addition, the site uses scripting for easy navigation, which is described in the nav.js JavaScript library.



The default.htm page in the Shopping Cart Example folder

Items that are placed in the Shopping Cart are added to the OrderedItems.fp5 database. Information that the web user enters on the customerid.htm page is added to a new record in the Customers.fp5 database. Items that are finally ordered (when the web user proceeds to Checkout and clicks Continue) are added to the Orders.fp5 database. The status of these orders remain open until the web user enters a Knitting Factory account number and expiration date.

All format files in these examples are well-commented, explaining how the CDML tags are used in every step.

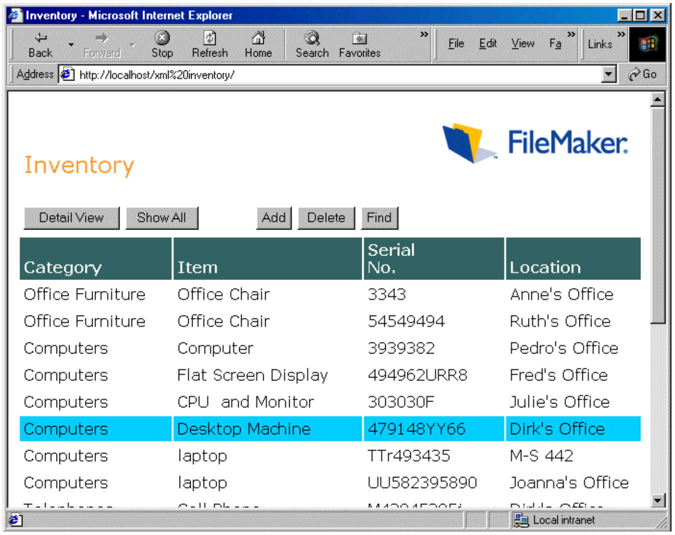
Chapter 5
Using FileMaker Pro XML to deliver your data

With the enhanced Web Companion, FileMaker Pro 5.5 and FileMaker Pro 5.5 Unlimited can deliver data from your database to the Web in Extensible Markup Language (XML) format. In the same way that HTML has become the standard display language for communication on the World Wide Web, XML promises to become the standard language for structured data interchange.

In XML format, FileMaker Pro data can be populated in your web page programmatically instead of downloading statically from the server. This gives you more flexibility and a more web-like application that allows your web users to interact with the data after it has been downloaded. This also allows the web server to handle more requests as more processing is done by the browser on web users' machines.

About the XML examples

The *FMWSC and Tools* CD includes an XML example that demonstrates how you can publish your database on the Web using XML and Dynamic HTML (including JavaScript). This XML example is designed to work in the Microsoft Internet Explorer 5 for Windows web browser. For step-by-step instructions, see "Looking at the XML Inventory example" on page 5-17.



http://localhost/xml%20inventory/

Inventory

Detail View | Show All | Add | Delete | Find

Category	Item	Serial No.	Location
Office Furniture	Office Chair	3343	Anne's Office
Office Furniture	Office Chair	54549494	Ruth's Office
Computers	Computer	3939382	Pedro's Office
Computers	Flat Screen Display	494962URR8	Fred's Office
Computers	CPU and Monitor	303030F	Julie's Office
Computers	Desktop Machine	479148YY66	Dirk's Office
Computers	laptop	TTr493435	M-S 442
Computers	laptop	UU582395890	Joanna's Office
Telephones	Cell Phone	[illegible]	Dirk's Office

List View of the Inventory.fp5 database in the XML Inventory example

For examples showing the differences between using stylesheets or scripting (Dynamic HTML) with your FileMaker Pro XML documents, see "Comparing CSS, XSLT, and JavaScript" on page 5-11.

For general information on XML (including a glossary of XML terms), additional examples that use XML, and links to XML resources, see the FileMaker, Inc. web site at www.filemaker.com. As a shortcut to the site, double-click FileMaker on the Web (included on the *FileMaker Pro* CD).

General process for custom web publishing using XML

Here’s a simple overview of the process for publishing your FileMaker Pro database on the Internet or an intranet using XML:

1. You send a FileMaker Pro CGI request (such as to find records in the database) to the Web Companion through an HTML form, an HREF link, or a script on your web page. The request can also be made by typing the URL in the web browser.

See “Generating FileMaker Pro CGI requests for an XML document” on page 5-8 and appendix A, “Valid names used in CGI requests for FileMaker XML data.”

2. The Web Companion generates an XML document containing the results of your request in XML format (for example, a found set of records from the database and an XML-stylesheet processing instruction) and returns it to your web browser.

See “Generating an XML document” next.

3. The web browser, with the help of an XML parser, applies any instructions that you’ve specified via a stylesheet and displays the data in HTML format.

See “Using style sheets with your XML document” on page 5-10.

Once the XML document is downloaded to your web browser, you can use stylesheets (such as CSS or XSL) to apply text formatting styles and object positioning, or scripting (such as JavaScript) to manipulate the data however you want. See “Comparing CSS, XSLT, and JavaScript” on page 5-11.

Generating an XML document

When you specify an XML format parameter in your FileMaker Pro CGI request, the Web Companion generates an XML document containing data from your database that is formatted by one of two types of XML grammars (or schemas).

One type (called FMPDSO) gives you more flexibility and control over individual elements and is ideally suited for use with cascading style sheets (CSS) or Extensible Stylesheet Language (XSL). The FMPDSO grammar can also be used with the Microsoft XML Data Source Object (DSO) in Internet Explorer 4.0 to publish read-only databases. (The Microsoft XML DSO lets you view but not update data in XML format.)

The other type of grammar (called FileMaker Extended XML or FMPXML) provides a broader, richer XML that defines FileMaker Pro layouts, fields, and value list information. These grammars can be combined with XSL documents or scripting (such as JavaScript) to publish dynamic databases on the Web.

All XML data generated by the Web Companion is well-formed and compliant with the XML 1.0 specification. The document type definitions (DTDs) for the grammars are provided in HTML documents for your convenience (included on the *FMWSC and Tools* CD).

FMWSC and Tools > XML > Documentation

Two of the grammars generated by the Web Companion are used for retrieving query results and a third is used for retrieving layout information. Depending on what you specify in your FileMaker Pro CGI request, the Web Companion will generate an XML document using one of these grammars:

- the FMPDSORESULT grammar
- the FMPXMLRESULT grammar
- the FMPXMLLAYOUT grammar

Each XML document contains a default XML namespace declaration for the grammar. (See “About XML namespaces” next.) You can also specify that the document contain an XML-stylesheet processing instruction. (See “Using style sheets with your XML document” on page 5-10.)

Use one of these grammars in your document or web page to display and work with FileMaker data in XML format.

Note XML data generated by the Web Companion is encoded using UTF-8 format (Unicode Transformation Format 8). For information, see “About UTF-8 encoded data” on page 5-8.

About XML namespaces

To avoid name collisions, unique XML namespaces help distinguish XML tags by the application they were designed for. For example, if your XML document contains two DATABASE elements, one for FileMaker Pro XML data and another for Oracle XML data, the namespaces will identify the DATABASE element for each.

The FileMaker Pro Web Companion generates a default namespace for each grammar. For example, for the FMPDSORESULT grammar, the following namespace is generated:

```
xmlns=“http://www.filemaker.com/fmpdsoresult”
```

About FileMaker Pro database error codes

The FileMaker Pro Web Companion generates an error code at the beginning of each grammar based on the current error status of the database. A value of zero (0) is returned for no error.

```
<ERRORCODE>0</ERRORCODE>
```

See appendix B, “FileMaker Pro values for error codes” for information.

Using the FMPDSORESULT grammar

When you specify “–dso_xml” as the format for a FileMaker Pro CGI request, the Web Companion will generate XML data based on a database-specific grammar that uses field names as element names. The FMPDSORESULT grammar is useful for publishing databases on web pages that are formatted with cascading style sheets or XSLT. (See “Comparing CSS, XSLT, and JavaScript” on page 5-11 for information.) The FMPDSORESULT grammar is compatible with the Microsoft XML Data Source Object used by Internet Explorer 4.0.

The Web Companion will also generate the document type definition for the grammar if you specify “–dso_xml_dtd” as the format. This is useful if you want an XML parser to validate the XML before your document goes to production.

Note Internet Explorer 4.0 directly supports XML with no additional software required. The XML can be displayed using dynamic data binding features available in the browser. This is accomplished with a Java applet that ships with Internet Explorer 4.0, which presents the XML as a Data Source Object (DSO) to the browser. With the DSO, the The Internet Explorer 4.0 browser exposes XML data to scripting languages such as JavaScript or VBScript via the Microsoft Document Object Model (DOM). Keep in mind that the Microsoft XML DSO applet does not provide a mechanism for updating the data, nor does it know anything about FileMaker Pro database layouts or value lists.

The following is an example of a Microsoft XML DSO applet tag that you might use in your web page to query FileMaker Pro for XML data using the FMPDSORESULT grammar—where the “url” parameter can be any valid FileMaker Pro CGI request containing a –format parameter equal to “–dso_xml” or “–dso_xml_dtd.” (See “Generating FileMaker Pro CGI requests for an XML document” on page 5-8 for a list of valid FileMaker CGI requests.)

```
<applet code=com.ms.xml.dso.XMLDSO.class width=0 height=0
id=xmldso MAYSCRIPT=true>
        <PARAM NAME="url" VALUE=fmpro?–db=PhoneList.fp5&
        –format=–dso_xml&–find=>
</applet>
```

Description of elements in the FMPDSORESULT grammar

Each ROW element in the generated FMPDSORESULT grammar contains a number of FIELD elements that correspond to the field names in the specified layout.

Spaces or single colons in field names are converted to underscores in the element names (for example, <FIRST_NAME>). Double colons in portal fields are converted to periods (for example, <PHONE.PHONE_NUMBER>). This is done because colons are reserved in XML for specifying namespaces and spaces are not allowed in XML element names.

For repeating and portal fields, each FIELD element will contain a DATA element that corresponds to each repetition or portal record.

Note The content of container fields in the database will be generated in the form of the relative URL used for retrieving the content instead of the actual content (such as an image).

To qualify the XML elements for the FileMaker Pro application, the names of all elements and attributes in this grammar are associated with the unique XML namespace http://www.filemaker.com/fmpdsoresult. This namespace is declared in the grammar as the default namespace.

The following is an example of XML data generated with the FMPDSORESULT grammar.

Example of XML data in the FMPDSORESULT grammar

```
<?xml version="1.0" encoding="UTF-8"?>
<FMPDSORESULT xmlns=“http://www.filemaker.com/fmpdsoresult”>
        <ERRORCODE>0</ERRORCODE>
        <DATABASE>PhoneList.fp5</DATABASE>
        <LAYOUT>Web Layout</LAYOUT>
        <ROW RECORDID=“3” MODID=“23”>
                <FIRST_NAME>John</FIRST_NAME>
                <LAST_NAME>Smith</LAST_NAME>
                <PHONE.PHONE_NUMBER>
                        <DATA>555-444-3333</DATA>
                        <DATA>555-222-9999</DATA>
                </PHONE.PHONE_NUMBER>
        </ROW>
        <ROW RECORDID=“6” MODID=“32”>
                <FIRST_NAME>Barbara</FIRST_NAME>
                <LAST_NAME>Jones</LAST_NAME>
                <PHONE.PHONE_NUMBER>
                        <DATA>555-666-7777</DATA>
                        <DATA>555-333-0000</DATA>
                        <DATA>555-111-7654</DATA>
                </PHONE.PHONE_NUMBER>
        </ROW>
</FMPDSORESULT>
```

Note If the –lay parameter is not specified in the FileMaker Pro CGI request, the LAYOUT element is empty and data for every field in the database is returned. (See “Generating FileMaker Pro CGI requests for an XML document” on page 5-8 for information.)

Using the FileMaker Pro Extended XML grammars

The FileMaker Pro Extended XML grammars contain additional information about field types, value lists and layouts that is not found in the FMPDSORESULT grammar. Use the FMPXMLRESULT and FMPXMLLAYOUT grammars if you require layout information or want the METADATA information provided by these grammars.

Note These grammars are not well suited for cascading style sheets with positioning. See “Using the FMPDSORESULT grammar” on page 5-3 if you want to use CSS with your XML data.

When you specify “–fmp_xml” as the format for a FileMaker Pro CGI request, the Web Companion will generate XML data using either the FMPXMLRESULT or FMPXMLLAYOUT grammar, depending on the request you specify in the CGI command:

- The Web Companion will generate the FMPXMLRESULT grammar when you specify –edit, –delete, –find, –new, –dbnames, –layoutnames, –scriptnames or –dbopen as the FileMaker CGI request.
- The Web Companion will generate the FMPXMLLAYOUT grammar when you specify –view as the FileMaker CGI request.

The Web Companion will also generate the document type definition for the grammar if you specify “–fmp_xml_dtd” as the format. This is useful if you want an XML parser to validate the XML before your document goes to production.

For a list of valid FileMaker CGI requests, see “Generating FileMaker Pro CGI requests for an XML document” on page 5-8.

Note When using XML grammars, you should do a case-insensitive compare for proper results.

Description of elements in the FMPXMLRESULT grammar

In the generated FMPXMLRESULT grammar, the DATABASE element contains attributes for the name of the database, the number of records in the database, the name of the layout that was used to generate the result set, and the format of dates and times in the XML document.

The DATEFORMAT attribute specifies the format of dates in the XML document.

Field	Full form	Short form
Year	yyyy (4 digits)	yy (2 digits)
Month	mm (2 digits)	M (1 or 2 digits)
Day	dd (2 digits)	d (1 or 2 digits)

The TIMEFORMAT attribute specifies the format of times in the XML document.

Field	Full form	Short form
Hour (1 – 12)	hh (2 digits)	h (1 or 2 digits)
Hour (1 – 24)	kk (2 digits)	k (1 or 2 digits)
Minute	mm	
Second	ss	
AM/PM	a	

The METADATA element contains one or more FIELD elements, each containing information for one of the fields/columns of the result set—including the name of the field as defined in the database, the field type, the Yes or No allowance for empty fields (EMPTYOK attribute) and the maximum number of repeating values (MAXREPEAT attribute). Valid values for field types are TEXT, NUMBER, DATE, TIME, and CONTAINER.

The RESULTSET element contains all of the ROW elements returned as the result of a query and an attribute for the total number of records found. Each ROW element contains the field/column data for one row in the result set—including the record ID for the row, the modification ID for the row, and the COL element containing the data for one field/column in the row (where multiple DATA elements represent one of the values in a repeating or portal field).

Note The content of container fields in the database will be generated in the form of the relative URL used for retrieving the content, instead of the actual content (such as an image).

To qualify the XML elements for the FileMaker Pro application, the names of all elements and attributes in this grammar are associated with the unique XML namespace http://www.filemaker.com/fmpxmlresult. This namespace is declared in the grammar as the default namespace.

The following is an example of XML data generated with the FMPXMLRESULT grammar.

Example of XML data in the FMPXMLRESULT grammar

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="yourstylesheet.xsl"?>
<FMPXMLRESULT xmlns="http://www.filemaker.com/fmpxmlresult">
    <ERRORCODE>0</ERRORCODE>
    <PRODUCT NAME="Web Companion" VERSION="5.0"
    BUILD="10/23/99"/>
    <DATABASE NAME="Employees.fp5" RECORDS="23"
    DATEFORMAT="MM/dd/yy" TIMEFORMAT="hh:mm:ss"
    LAYOUT="summary"/>
    <METADATA>
        <FIELD NAME="First Name" TYPE="TEXT"
        EMPTYOK="NO" MAXREPEAT=”1”/>
        <FIELD NAME="Last Name" TYPE="TEXT"
        EMPTYOK="NO" MAXREPEAT=”1”/>
        <FIELD NAME="Department" TYPE="TEXT"
        EMPTYOK="YES" MAXREPEAT=”1”/>
    </METADATA>
    <RESULTSET FOUND="5">
        <ROW RECORDID=”34” MODID=”47”>
            <COL>
                <DATA>Joe</DATA>
            </COL>
            <COL>
                <DATA>Smith</DATA>
            </COL>
            <COL>
                <DATA>Engineering</DATA>
            </COL>
        </ROW>
        <ROW RECORDID=”78” MODID=”89”>
            <COL>
                <DATA>Susan</DATA>
            </COL>
            <COL>
                <DATA>Jones</DATA>
            </COL>
            <COL>
                <DATA>Marketing</DATA>
            </COL>
        </ROW>
    </RESULTSET>
</FMPXMLRESULT>
```

The order of the COL elements corresponds with the order of the FIELD elements in the METADATA element—for example, where the “First Name”, “Last Name”, and then “Department” elements are listed in the METADATA, “Joe”, “Smith”, and then “Engineering” are listed in the same order in the RESULTSET ROW.

Note If the –lay parameter is not specified in the FileMaker Pro CGI request, the LAYOUT attribute in the DATABASE element is empty and data for every field in the database is returned. (See “Generating FileMaker Pro CGI requests for an XML document” on page 5-8 for information.)

Description of elements in the FMPXMLLAYOUT grammar

In the generated FMPXMLLAYOUT grammar, the LAYOUT element contains the name of the layout, the name of the database, and FIELD elements for each field found in the corresponding layout in the database. Each FIELD element describes the style type of the field, and contains the VALUELIST attribute for any associated value list of the field.

The VALUELISTS element contains one or more VALUELIST elements for each value list found in the layout—each including the name of the value list and a VALUE element for each value in the list.

To qualify the XML elements for the FileMaker Pro application, the names of all elements and attributes in this grammar are associated with the unique XML namespace http://www.filemaker.com/fmpxmllayout. This namespace is declared in the grammar as the default namespace.

The following is an example of XML data generated with the FMPXMLLAYOUT grammar.

Example of XML data in the FMPXMLLAYOUT grammar

```
<?xml version="1.0" encoding="UTF-8"?>
<FMPXMLLAYOUT xmlns="http://www.filemaker.com/fmpxmllayout">
    <ERRORCODE>0</ERRORCODE>
    <PRODUCT NAME="Web Companion" VERSION="5.0v1"
    BUILD="10/24/99"/>
    <LAYOUT NAME="Web Layout" DATABASE="employees.fp5">
        <FIELD NAME="First Name">
            <STYLE TYPE="EDITTEXT" VALUELIST="" />
        </FIELD>
        <FIELD NAME="Last Name">
            <STYLE TYPE="EDITTEXT" VALUELIST="" />
        </FIELD>
        <FIELD NAME="Department">
            <STYLE TYPE="POPUPMENU"
            VALUELIST="Departments" />
        </FIELD>
    </LAYOUT>
    <VALUELISTS>
        <VALUELIST NAME="Departments">
            <VALUE>Engineering</VALUE>
            <VALUE>Marketing</VALUE>
        </VALUELIST>
    </VALUELISTS>
</FMPXMLLAYOUT>
```

About UTF-8 encoded data

All XML data generated by the Web Companion is encoded in UTF-8 (Unicode Transformation 8 Bit) format. This format compresses data from the standard Unicode format of 16 bits to 8 bits for ASCII characters. XML parsers are required to support Unicode and UTF-8 encoding.

UTF-8 encoding includes direct representations of most of the characters used in English using values of 0-127 for the standard ASCII set of characters, and provides multibyte encodings for Unicode characters with higher values. UTF-8 encoded data is compressed almost in half (lower ASCII characters are compressed from 2 bytes to 1 byte), which helps data download faster.

Note Because your XML data is UTF-8 encoded, some upper ASCII characters will be represented by 2 or 3 characters in the text editor—they will appear as single characters only in the XML parser or browser.

The UTF-8 encoding format includes the following features:

- All ASCII characters are one-byte UTF-8 characters. A legal ASCII string is a legal UTF-8 string.
- Any non-ASCII character (i.e., any character with the high-order bit set) is part of a multibyte character.
- The first byte of any UTF-8 character indicates the number of additional bytes in the character.
- The first byte of a multibyte character is easily distinguished from the subsequent bytes. Thus, it is easy to locate the start of a character from an arbitrary position in a data stream.
- It is easy to convert between UTF-8 and Unicode.
- The UTF-8 encoding is relatively compact. For text with a large percentage of ASCII characters, it is more compact than Unicode. In the worst case, a UTF-8 string is only 50% larger than the corresponding Unicode string.

Generating FileMaker Pro CGI requests for an XML document

You use FileMaker Pro CGI (Common Gateway Interface) commands to generate requests for XML data from your database.

For example, to generate a –find request to display all employees from a database, web users might click on a link containing the following FileMaker Pro CGI command:

```
FMPro?–db=employees.fp5&–format= –dso_xml&
–styletype=text/css&–stylehref=stylesheet.css&–find
```

Request and parameter names

The following tables list the request and parameter names in name/value pairs you can use in a FileMaker Pro CGI command when requesting data in XML format.

For more information and examples, see appendix A, “Valid names used in FileMaker CGI requests for XML data.”

Use this request name	To generate this request
–new	New record
–edit	Edit record
–delete	Delete record
–find	Find record(s)
–findall	Find all records
–findany	find a random record
–view	View layout info (in FMPXMLLAYOUT grammar)
–dbnames	Retrieve names of all open and web-shared databases
–layoutnames	Retrieve names of all available layouts for a specified open, web-shared database

Use this request name	To generate this request
–scriptnames	Retrieve names of all available scripts for a specified open, web-shared database
–dbopen	Open a database that’s in the Web folder with Remote Administration enabled
–dbclose	Close a database that’s in the Web folder with Remote Administration enabled

Use these parameter names	To go with these requests
–db (database name)	Required for all requests except –dbnames
–lay (layout name)	Required for –view, and with –edit or –new requests for data in related fields and portals. Optional for –find, –findall
–format	Required for all requests. (Use one of these formats: –dso_xml, –dso_xml_dtd, –fmp_xml, or –fmp_xml_dtd)
–recid (record I.D.)	Required for –edit and –delete. Optional for –find
–modid (modification I.D.)	Optional for –edit
–lop (logical operator)	Optional for –find
–op (operator)	Optional for –find
–max (maximum records)	Optional for –find
–skip (skip records)	Optional for –find
–sortorder (sort order)	Optional for –find, –findall
–sortfield (sort field)	Optional for –find, –findall
–script (perform script)	Optional for –find, –findall
–script.prefind (perform script before –find)	Optional for –find, –findall
–script.presort (perform script before sort)	Optional for –find, –findall
–styletype (stylesheet type)	Optional for all requests
–stylehref (stylesheet HREF)	Optional for all requests
–password	Optional for –dbopen requests. Specifies the database’s password.
field name (no hyphen)	At least one field name is required for –new and –edit. Optional for –find. See “field name (Name of specific field)” on page A-10 for more information.

Note The -max parameter now returns 0 if the request returns no records.

Requests for adding records to a portal

When you make an –edit request or a –new request that includes data for a portal of related database records, you must specify the layout and the relationship name for the related database.

Note You can only add one record at a time to a portal, and therefore must make separate –new requests to add more rows to the portal.

The following is an example of a –new request for adding a record to a portal, where “Address::” is the name of the database relationship, and “City.0” is the related field name in the portal:

```
FMPro?–db=employees.fp5&–lay=LayoutOne&FirstName=Sam
&LastName= Smith&Address::City.0=Seattle&–format= –fmp_xml&–new
```

Requests for editing multiple records in a portal

You only need to make one –edit request to edit multiple records in a portal. You specify each row (or record) in the portal by adding a period and a consecutive number (starting with number 1) to the end of the related field name.

The following is an example of an –edit request for editing records in a portal, where “Address::” is the name of the relationship, “City.1” is the first row in the portal, and “City.2” is the second row in the portal:

```
FMPro?–db=employees.fp5&–lay=LayoutOne&recid=11&
FirstName=Sam&LastName=Smith&Address::City.1=Seattle
&Address::City.2=New York&–format= –fmp_xml&–edit
```

The following is an example of another –edit request for editing records in a portal, in an HTML form:

```
<FORM ACTION="fmpro" METHOD="POST">
    <INPUT TYPE="HIDDEN" NAME="-db" VALUE="employees.fp5">
    <INPUT TYPE="HIDDEN" NAME="-lay" VALUE="LayoutOne">
    <INPUT TYPE="HIDDEN" NAME="-format" VALUE="-fmp_xml">
    <INPUT TYPE="HIDDEN" NAME="-recid" VALUE="11">
    <INPUT TYPE="TEXT" NAME="FirstName" VALUE="Joe">
    <INPUT TYPE="TEXT" NAME="LastName" VALUE="Smith">
    <INPUT TYPE="TEXT" NAME="Address::City.1" VALUE="San
    Jose">
    <INPUT TYPE="TEXT" NAME="Address::City.2" VALUE="Santa
    Clara">
    <INPUT TYPE="SUBMIT" NAME="-edit" VALUE="Edit Record">
</FORM>
```

Using style sheets with your XML document

The Web Companion will generate an XML-stylesheet processing instruction with each grammar if the FileMaker CGI request includes the –styletype and –stylehref parameters. This allows you to use cascading style sheets (CSS) or Extensible Stylesheet Language (XSL) documents for displaying your XML document.

The –styletype parameter is used for setting the value of the type attribute (type=text/css or type=text/xsl).

The –stylehref parameter is used for setting the value of the HREF attribute (href=document.css or href=document.xsl).

Here is an example of what a possible FileMaker CGI command might look like:

```
http://localhost/fmpro?–db=employees.fp5&–format= –fmp_xml&–find=&
–styletype=text/xsl&–stylehref=document.xsl
```

Based on this command, the Web Companion will include the following processing instruction in the XML document:

```
<?xml-stylesheet type=“text/xsl” href=“document.xsl”?>
```

The following text is an example of a possible XSL document used with the FMPXMLRESULT grammar. In this example, the XSL document converts the XML document into an HTML document by inserting HTML tags. It builds an HTML table that contains a header row for all the field names from the METADATA element in the FMPXMLRESULT grammar, and table rows for all the field data in the ROW elements of the RESULTSET.

Note This is an example of XSLT that was written to work with Internet Explorer 5.0 for Windows, not with other browsers using later versions of XSLT.

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl"
xmlns:HTML="http://www.w3.org/Profiles/XHTML-transitional">
    <xsl:template>
        <xsl:apply-templates/>
    </xsl:template>


    <xsl:template match="text()">
        <xsl:value-of/>
    </xsl:template>

    <xsl:template match="/">
        <HTML>
            <BODY>
                <CENTER>
                <TABLE BORDER="0">
                <xsl:apply-templates/>
                </TABLE>
                </CENTER>
            </BODY>
        </HTML>
    </xsl:template>

    <xsl:template match="ERRORCODE">
    </xsl:template>

    <xsl:template match="METADATA">
        <TR>
            <xsl:apply-templates/>
        </TR>
    </xsl:template>

    <xsl:template match="FIELD">
        <TD ALIGN="CENTER" BGCOLOR="#336666">
            <FONT FACE="Verdana"
            COLOR="#FFFFFF"><B><xsl:value-of
            select="@NAME"/></B></FONT>
        </TD>
    </xsl:template>

    <xsl:template match="RESULTSET/ROW">
        <TR>
            <xsl:apply-templates/>
        </TR>
    </xsl:template>

    <xsl:template match="COL">
        <TD BGCOLOR="#00CCFF">
            <xsl:apply-templates/>
        </TD>
    </xsl:template>

    <xsl:template match="DATA">
        <P><xsl:value-of/></P>
    </xsl:template>

</xsl:stylesheet>
```

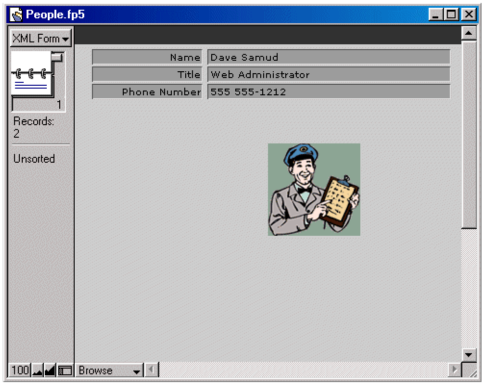
Comparing CSS, XSLT, and JavaScript

The *FMWSC and Tools* CD includes three simple examples that demonstrate the differences between using cascading style sheets (CSS), Extensible Stylesheet Language–Transformations (XSLT), and JavaScript scripting language with your XML documents.

These examples are included in the XML folder:

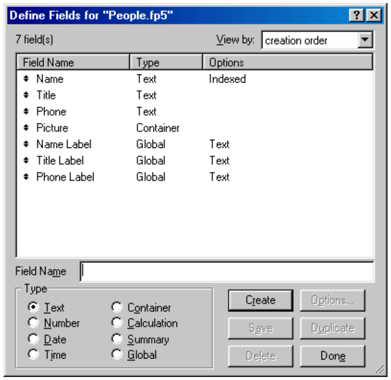
FMWSC and Tools > XML > Simple Examples

All three examples use a simple database named People.fp5.



Name Dave Samud
Title Web Administrator
Phone Number 555 555-1212

The People.fp5 database contains seven fields — three text fields for data, three global fields for field labels, and one container field for pictures.



7 field(s)   View by: creation order

Field Name	Type	Options
Name	Text	Indexed
Title	Text	
Phone	Text	
Picture	Container	
Name Label	Global	Text
Title Label	Global	Text
Phone Label	Global	Text

Field Name

Type
● Text ○ Container
○ Number ○ Calculation
○ Date ○ Summary
○ Time ○ Global

The seven fields in the People.fp5 database used for these examples

These three examples were designed to be viewed in the Internet Explorer 5.0 for Windows web browser. For information on new browsers that can be used to view the examples, double-click FileMaker on Web (included on the *FMWSC and Tools* CD) to go to the FileMaker product support pages in your browser.

To view the examples:

1. Place a copy of the Simple Examples folder and its files in the Web folder inside the FileMaker Pro 5.5 application folder on your hard disk.

FileMaker Pro 5.5 > Web > Simple Examples

Note For security, when actually publishing a database on the Web, you should not keep the database file in the Web folder unless you plan to administer it remotely. See “Opening password-protected databases remotely” on page 3-18 for more information.

2. In FileMaker Pro, open the People.fp5 database and make sure it’s shared via the Web Companion.

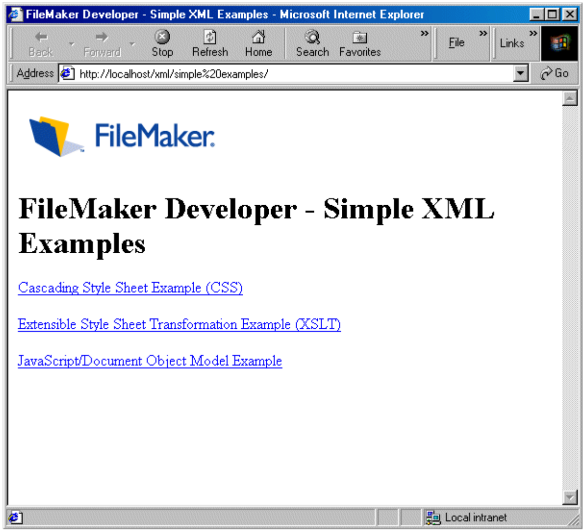
See “Enabling the Web Companion” on page 3-3 and “Sharing the database via the Web” on page 3-5 for information.

3. In your web browser, type `localhost` (or your computer’s IP address) followed by `/simple examples/` and press Enter.

http://localhost/simple examples/

http://17.17.17.17/simple examples/

For information on setting up your computer as the localhost, see “Testing your site without a network connection” on page 3-18.


FileMaker Developer - Simple XML Examples - Microsoft Internet Explorer
Back
Forward
Stop
Refresh
Home
Search
Favorites
File
Links
Address
http://localhost/xml/simple%20examples/
Go
FileMaker
Local intranet

FileMaker Developer - Simple XML Examples

Cascading Style Sheet Example (CSS)

Extensible Style Sheet Transformation Example (XSLT)

JavaScript/Document Object Model Example

4. On the Default.htm page, click the links to the CSS, XSL, and Scripting/DOM examples, or view the source to see the FileMaker Pro CGI requests for the links.

See “Generating FileMaker Pro CGI requests for an XML document” on page 5-8 for information.

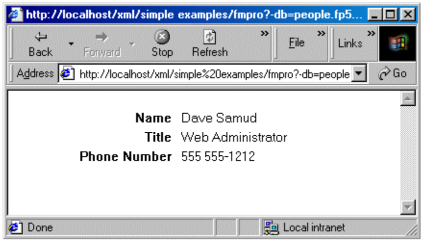
Cascading style sheets (CSS) example

Cascading style sheets (CSS) documents format the text style (font, size, color, etc.) and positioning of your data in an XML document.

A cascading style sheet can only be applied to the existing data in an XML document—it cannot be used to transform the data (such as transforming a URL for a container field into its corresponding image), or to add additional information (such as labels for each field) to the XML document.

This example demonstrates the use of a CSS document with an XML document. The following CGI command was used to generate the FMPDSORESULT grammar for the People.fp5 database and to apply the People_form.css stylesheet to the generated XML data:

```
fmpro?–db=people.fp5&–lay=xml form&–format=dso_xml&
–styletype=text/css&–stylehref=people_form.css&–max=1&–find=
```


http://localhost/xml/simple examples/fmpro?-db=people.fp5...
Back
Forward
Stop
Refresh
File
Links
Address
http://localhost/xml/simple%20examples/fmpro?-db=people
Go
Done
Local intranet

Name Dave Samud
Title Web Administrator
Phone Number 555 555-1212

To simulate field labels, the CSS document applies boldface and right-alignment to data in the three global fields

The picture in the container field is not displayed in this example because the CSS document can only apply styles to the existing data (the relative URL to the image). It’s not possible to transform the URL into its corresponding image using cascading style sheets.

The following text is the people_form.css stylesheet, which adds formatting and positioning to the XML:

```
/*
 * Since FMPDSORESULT is the root element of the XML document,
 * any formatting applied to this element will cascade down to all
 * other elements in the document. This is a good place to set the
 * default formatting options for your page.
 */
FMPDSORESULT
{
        font-family: sans-serif;
        font-size: 10pt;
}
/*
 * Don't display the data for these elements.
 */
```

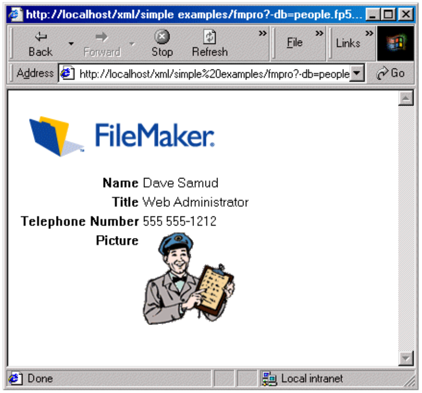
```
ERRORCODE, DATABASE, LAYOUT, Picture
{
        visibility: hidden;
}
/*
 * Since it is not possible to add additional text to the output
 * of the XML document via CSS, we must rely on the XML document to
 * provide the labels when displaying the field data. In this case,
 * the labels are from global fields defined in the database.
 */
Name_Label
{
        position: absolute;
        left: 20px;
        top: 20px;
        width: 150px;
        text-align: right;
        font-weight: bold;
}
Title_Label
{
        position: absolute;
        left: 20px;
        top: 40px;
        width: 150px;
        text-align: right;
        font-weight: bold;
}
Phone_Label
{
        position: absolute;
        left: 20px;
        top: 60px;
        width: 150px;
        text-align: right;
        font-weight: bold;
}
/*
 * The following styles are applied to the actual field data.
 */
Name
{
        position: absolute;
        left: 180px;
        top: 20px;
}
Title
{
        position: absolute;
        left: 180px;
        top: 40px;
}
Phone
{
        position: absolute;
        left: 180px;
        top: 60px;
}
```

Extensible Stylesheet Language–Transformations (XSLT) example

Extensible Stylesheet Language–Transformations (XSLT) is a language for transforming XML documents into other XML documents. XSLT is part of the overall XSL specification, which also includes an XML vocabulary for formatting the transformed XML document (such as applying text styles).

This example demonstrates the use of an XSL document with an XML document. The following CGI command was used to generate the FMPDSORESULT grammar for the People.fp5 database and to apply the People_form.xsl stylesheet to the generated XML data:

```
fmpro?–db=people.fp5&–lay=xml form&–format=dso_xml&
–styletype=text/xsl&–stylehref=people_form.xsl&–max=1&–find=
```


http://localhost/xml/simple examples/fmpro?-db=people.fp5...
Back
Forward
Stop
Refresh
File
Links
Address
http://localhost/xml/simple%20examples/fmpro?-db=people
Go
FileMaker
Done
Local intranet

Name Dave Samud
Title Web Administrator
Telephone Number 555 555-1212
Picture

XSL lets you display images in container fields, format the data, and add field labels and a logo to the XML document

The three global fields for field labels in the People.fp5 database are not necessary when you’re using an XSL document. You can use XSLT to add labels after the XML document has been generated by FileMaker Pro.

Note You can also include scripting (such as JavaScript) in your XSL document. See “JavaScript scripting language example” next.

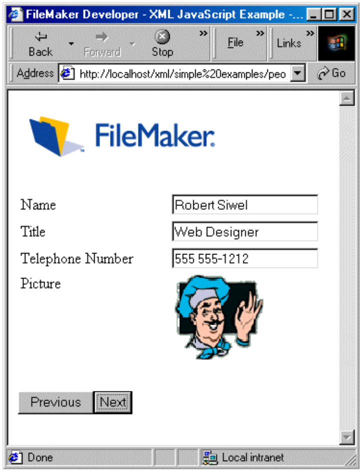
The following text is the people_form.xsl stylesheet, which adds labels and transforms the URL in the Picture field into an image:

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl"
xmlns:fm="http://www.filemaker.com/fmpdsoresult">
    <!--
    The following two templates have been added for IE5 Windows,
    since that browser's XSL processor does not provide defaults for
    them.
    -->
    <xsl:template><xsl:apply-templates/></xsl:template>
    <xsl:template match="text()"><xsl:value-of select="."/></
    xsl:template
    <xsl:template match="fm:FMPDSORESULT">
        <html>
            <head>
                <style type="text/css">
                    table {font-family: sans-serif; font-size:
                    10pt; }
                    td.label { text-align: right; vertical-align:
                    text-top; font-weight: bold; }
                </style>
            </head>
            <body>
                <img src="/xml/simple examples/FileMaker.gif"
                />
                <xsl:apply-templates/>
            </body>
        </html>
    </xsl:template>
    <xsl:template match="fm:ROW">
        <table>
            <tr>
                <td class="label">Name</td>
                <td><xsl:value-of select="fm:Name"/></td>
            </tr>
            <tr>
                <td class="label">Title</td>
                <td><xsl:value-of select="fm:Title"/></td>
            </tr>
            <tr>
                <td class="label">Telephone Number</td>
                <td><xsl:value-of select="fm:Phone"/></td>
            </tr>
            <tr>
                <td class="label">Picture</td>
                <td><xsl:element name="img"><xsl:attribute
                name="src"><xsl:value-of select="fm:Picture"/>
                </xsl:attribute></xsl:element></td>
            </tr>
        </table>
    </xsl:template>
    <!--
        Don't display anything for the following elements.
    -->
    <xsl:template
    match="fm:ERRORCODE|fm:DATABASE|fm:LAYOUT">
    </xsl:template>
</xsl:stylesheet>
```

JavaScript scripting language example

Using HTML and a scripting language with your XML document can allow your web users to interact with the database after it has been downloaded. For example, a simple onClick scripting event handler can allow web users to click a button and see different records in the database.

This example demonstrates the use of the JavaScript scripting language with an XML document to publish the People.fp5 database on a web page. It starts with an HTML file, named People_form.htm, that 1) references the JavaScript library, FMP.js, 2) contains a simple HTML form with a table for the field and picture rows, and 3) builds and executes a CGI command to FileMaker Pro to find and download all the records in the People.fp5 database.



http://localhost/xml/simple%20examples/peo

Name: Robert Siwel
Title: Web Designer
Telephone Number: 555 555-1212
Picture

With JavaScript, you can manipulate the data in the XML document to display different records in the database—after the data has been downloaded

Note The FMP.js JavaScript library was created for the XML Inventory example, described next, which is a more complex and sophisticated example of the use of JavaScript and the W3C Document Object Model with your XML documents.

Once the people_form.htm page is loaded in the browser, the onLoad event handler performs the “initialize” function, creating an ActiveXObject and building the FMPFindRequest:

```
function initialize ( )
{
    var xmlDocument = new ActiveXObject (“Microsoft.XMLDOM”);
    xmlDocument.async = false;
    var findRequest = new FMPFindRequest (“people.fp5”, null, AND,
    false);
    if (xmlDocument.load(findRequest.getRelativeURL( ) ) )
    {
        foundSet = new FMPFoundSet (xmlDocument);
        populateFields( );
    }
    else
        alert (“Error retrieving records.”);
}
```

The xmlDocument and findRequest variables are referenced from the FMP.js JavaScript library. The values in the new FMPFindRequest are people for the name of the database, null for no layout, AND for the find criteria’s logical operator, and false for no XML DTD to be generated as a result of the request.

In the People_form.htm page, the nextRecord and previousRecord functions are assigned to the onClick event handler of the Next and Previous buttons respectively, so that when web users click the buttons, the function is performed and the fields are re-populated with new data.

```
function nextRecord( )
{
        if (foundSet.nextRecord( ))
                populateFields( );
}
function previousRecord( )
{
        if (foundSet.previousRecord( ))
                populateFields( );
}
function populateFields( )
{

        document.getElementsByName(“Name”) . item(0) . value =
                foundSet.getFieldByName (“Name”);
        document.getElementsByName(“Title”) . item(0) . value =
                foundSet.getFieldByName (“Title”);
        document.getElementsByName(“Phone”) . item(0) . value =
                foundSet.getFieldByName (“Phone”);
        document.getElementsByName(“Picture”) . item(0) . src =
                foundSet.getFieldByName (“Picture”);
}
```

Looking at the XML Inventory example

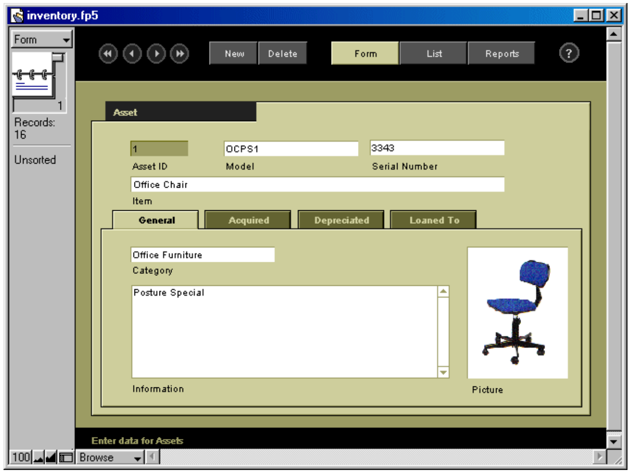
The XML Inventory example is a demonstration of a web-published database that uses the FMPXMLRESULT and FMPXMLLAYOUT grammars and Dynamic HTML (including JavaScript and the W3C Document Object Model) to display the XML data on the Web.

This example was designed to be viewed in the Internet Explorer 5.0 for Windows web browser. For information on new browsers that can be used to view the example, double-click FileMaker on the Web (included on the *Filemaker Pro* CD) to go to the FileMaker product support pages in your browser.

The example uses an inventory database for office equipment and provides two methods for displaying records from the database—in a list or in a detailed view of each record. Web site visitors can switch between List View and Detail View, and add, edit, delete, or find records in the database.

The XML Inventory example includes:

- a text file containing the JavaScript library used for this demonstration, named FMP.js
- the Inventory.fp5 database
- HTML files for viewing, searching, and adding records to the database



Asset ID: 1
Model: OCPS1
Serial Number: 3343
Item: Office Chair
Category: Office Furniture
Information: Posture Special

The Inventory.fp5 database used in the XML Inventory example

To view the XML Inventory example:

1. If necessary, copy the Inventory example files from the *FMWSC and Tools* CD onto your hard disk.

FMWSC and Tools > XML > Inventory Example

2. Place a copy of the Inventory Example folder and its files in the Web folder inside the FileMaker Pro 5.5 application folder on your hard disk.

FileMaker Pro 5.5 > Web > Inventory Example

Note For security, when actually publishing a database on the Web, you should not keep the database file in the Web folder unless you plan to administer it remotely. See "Opening password-protected databases remotely" on page 3-18 for information.

3. In FileMaker Pro, open the Inventory.fp5 database and make sure it's shared via the Web Companion.

See "Enabling the Web Companion" on page 3-3 and "Sharing the database via the Web" on page 3-5 for information.

4. In your web browser, type `localhost` (or your computer's IP address) followed by `/inventory example/` and press Enter.

http://localhost/inventory example/

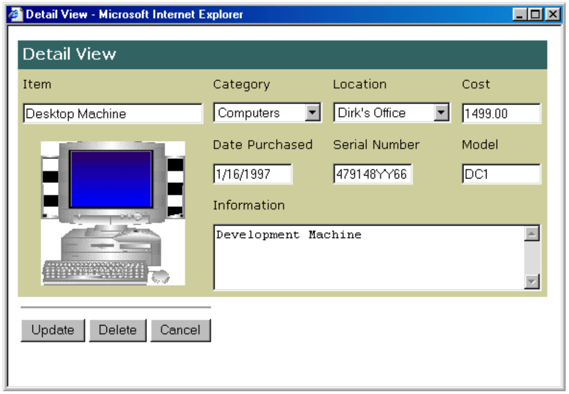
http://17.17.17.17/inventory example/

For information on setting up your computer as the localhost, see "Testing your site without a network connection" on page 3-18.

The XML Inventory example opens in List View in the browser.

5. Select a record and click the Detail View button.

The currently selected record displays in a separate Detail View window. It is also possible to open the Detail View window by double-clicking a record item in the list. To close the Detail View window, click the Cancel button. To apply edits made to a record in Detail View, click the Update button.

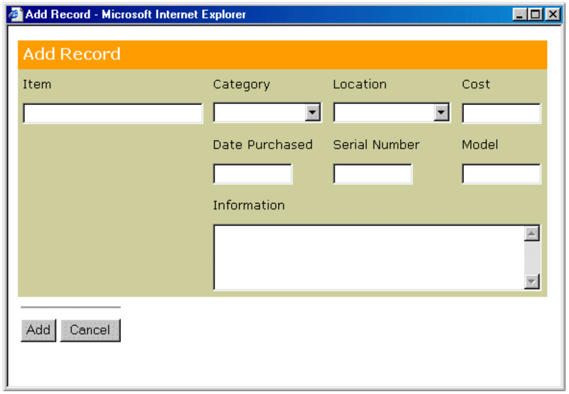


Item: Desktop Machine
Category: Computers
Location: Dirk's Office
Cost: 1499.00
Date Purchased: 1/16/1997
Serial Number: 479148YY66
Model: DC1
Information: Development Machine

Detail View of a selected record in the browser

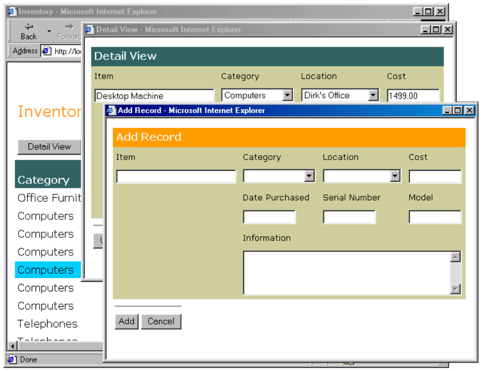
6. To add a record, click the Add button in the List View window.

A separate Add Record window opens.



7. Enter data for the new record and click Add to add it to the bottom of the list.

It is possible to have both the Detail View and Add Record windows open at the same time. The Detail View and Add Record windows will automatically close when a visitor clicks the Back or Forward button in the browser.

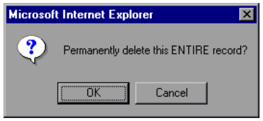


Desktop Machine | Computers | Dirk's Office | 1499.00

Detail View and Add Record windows open at the same time

8. To delete a record, click the Delete button in the Detail View or List View window.

A confirmation dialog box provides options to cancel or delete the currently selected record.

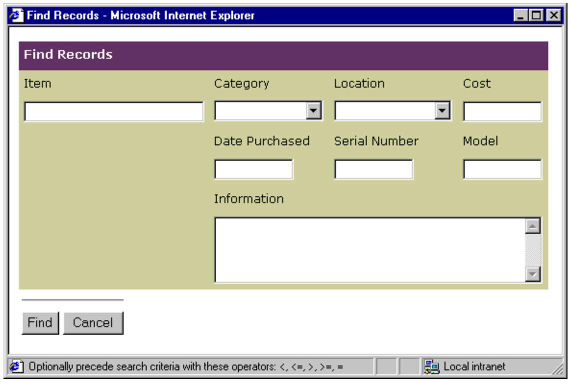


Permanently delete this ENTIRE record?

9. Click Cancel.

10. To search for data, click the Find button in the List View window.

A detail view of an empty record opens in a separate window with a blinking insertion point in the Item field. When web users click inside the Item, Cost, Date Purchased, Serial Number, or Notes text boxes, a help string appears in the status bar at the bottom of the window. The default operator for a search is “begins with” or web users can type one of the <, <=, >, >=, or = operator symbols in these boxes.

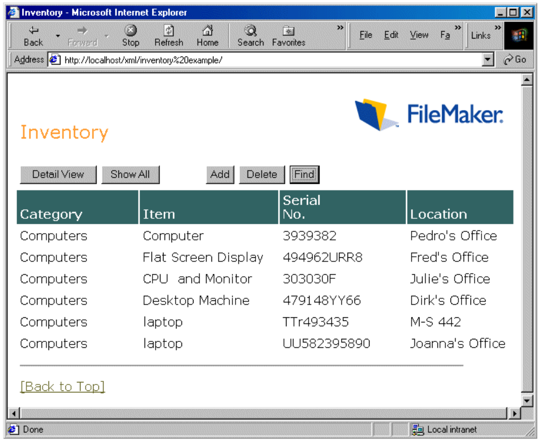


Optionally precede search criteria with these operators: <, <=, >, >=, =

Click in text boxes to display data entry instructions at bottom of window

11. Click in the Cost text box and type `>=2000`. Then click Find.

Any record containing the amount of $2,000 or greater in the Cost field is found and displayed in List View.



Inventory

Category	Item	Serial No.	Location
Computers	Computer	3939382	Pedro's Office
Computers	Flat Screen Display	494962URR8	Fred's Office
Computers	CPU and Monitor	303030F	Julie's Office
Computers	Desktop Machine	479148YY66	Dirk's Office
Computers	laptop	TTr493435	M-S 442
Computers	laptop	UU582395890	Joanna's Office

[Back to Top]

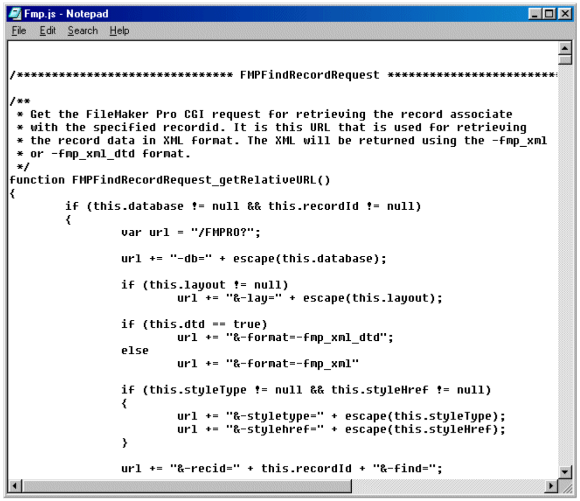
The found set of records is displayed in the List View window

12. In List View, click Show All to display all the records in the database, including any records you added.

If you want, you can examine the XML Inventory example's JavaScript library as you view the source for the example pages.

1. Start the Microsoft Notepad application (or a similar text editor), and choose File menu > Open. Locate and open the FMP.js file. (If necessary, choose All Files (*.*) as the Type.)

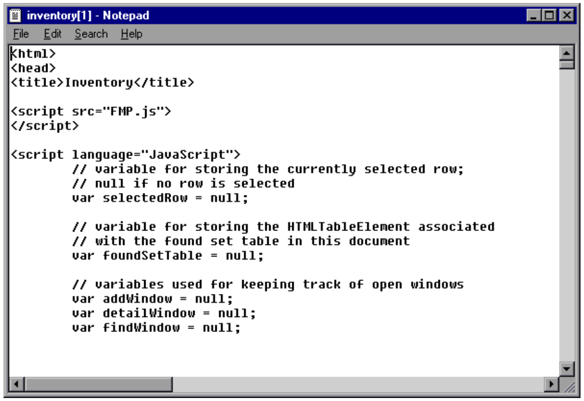
FileMaker Pro 5.5 > Web > Inventory Example > FMP.js



```
/******************************** FMPFindRecordRequest *************************

/**
 * Get the FileMaker Pro CGI request for retrieving the record associate
 * with the specified recordid. It is this URL that is used for retrieving
 * the record data in XML format. The XML will be returned using the -fmp_xml
 * or -fmp_xml_dtd format.
 */
function FMPFindRecordRequest_getRelativeURL()
{
        if (this.database != null && this.recordId != null)
        {
                var url = "/FMPRO?";

                url += "-db=" + escape(this.database);

                if (this.layout != null)
                        url += "&-lay=" + escape(this.layout);

                if (this.dtd == true)
                        url += "&-format=-fmp_xml_dtd";
                else
                        url += "&-format=-fmp_xml"

                if (this.styleType != null && this.styleHref != null)
                {
                        url += "&-styletype=" + escape(this.styleType);
                        url += "&-stylehref=" + escape(this.styleHref);
                }

                url += "&-recid=" + this.recordId + "&-find=";
```

FMP.js text file containing the JavaScript library for this example

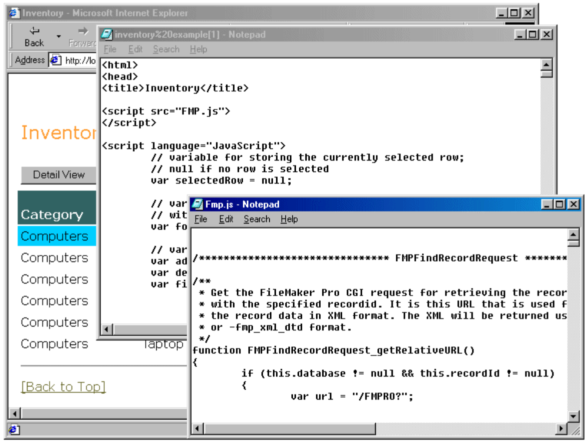
2. In the browser window, choose View menu > Source to see the source markup for the example pages in the Notepad application.



```
<html>
<head>
<title>Inventory</title>

<script src="FMP.js">
</script>

<script language="JavaScript">
        // variable for storing the currently selected row;
        // null if no row is selected
        var selectedRow = null;

        // variable for storing the HTMLTableElement associated
        // with the found set table in this document
        var foundSetTable = null;

        // variables used for keeping track of open windows
        var addWindow = null;
        var detailWindow = null;
        var findWindow = null;
```

Source markup of the Default.htm page (List View)

The scripts and functions are well commented in the source and in the FMP.js library, providing information about each step. It's useful to arrange the windows so you can switch between the example pages in the browser, the source, and the script library.



```
<html>
<head>
<title>Inventory</title>

<script src="FMP.js">
</script>

<script language="JavaScript">
        // variable for storing the currently selected row;
        // null if no row is selected
        var selectedRow = null;

        // var
        // wit
        var fo

        // var
        var ad
        var de
        var fi
```

```
/******************************* FMPFindRecordRequest *******

/**
 * Get the FileMaker Pro CGI request for retrieving the recor
 * with the specified recordid. It is this URL that is used f
 * the record data in XML format. The XML will be returned us
 * or -fmp_xml_dtd format.
 */
function FMPFindRecordRequest_getRelativeURL()
{
        if (this.database != null && this.recordId != null)
        {
                var url = "/FMPRO?";
```

Tip Use code snippets from the FMP.js JavaScript library for scripting your own web sites.

For more information about delivering your FileMaker Pro data using XML, double-click FileMaker on the Web (included on the *FileMaker Pro* CD).

This page intentionally left blank.

Chapter 6
Using Java and JDBC to deliver your data

If you're a Java programmer, you can use the FileMaker JDBC Driver with any Rapid Application Development (RAD) tool to visually create your FileMaker Pro database-aware Java application or applet.

The FileMaker JDBC Driver lets you directly access FileMaker Pro data using a RAD tool as you're building your code. Then, the Java application or applet that uses the FileMaker JDBC Driver can access FileMaker Pro data via the Web Companion.

About the JDBC examples

FileMaker Pro 5.5 Unlimited provides three examples of Java applications that use the FileMaker JDBC Driver to connect to a database. One example is a development-tool-independent Java application that was created using the basic Java classes and Sun Microsystems' Swing 1.1.1. The other two examples are Java front ends created with the development tools Corel's (Borland/Inprise) JBuilder 3.0 Professional for Windows and Symantec's Visual Cafe 4.0 Expert Edition for Windows.

For step-by-step instructions, see:

- "Example 1: Looking at the FileMaker Pro Explorer application" on page 6-8
- "Example 2: Creating the JBuilder Inventory application" on page 6-11
- "Example 3: Creating the Visual Cafe Inventory application" on page 6-14

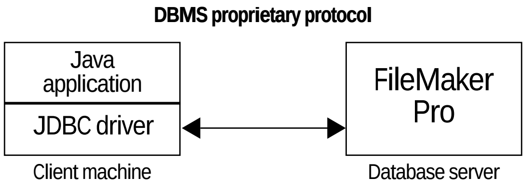
The *FMWSC and Tools* CD also includes the proprietary FileMaker Java classes and examples of Java applets that use them. For information, see "Using the FileMaker Java classes" on page 6-17.

For additional information and examples that use Java and JDBC for general data interchange or for publishing FileMaker Pro data on the Web, see the product support pages on the FileMaker, Inc. web site at www.filemaker.com. As a shortcut to the site, double-click FileMaker on the Web (included on the *FileMaker Pro* CD).

About JDBC

JDBC is a Java API for executing Structured Query Language (SQL) statements, the standard language for accessing relational databases. JDBC is a trademarked name and not an acronym—although it is thought of as standing for Java Database Connectivity because it is the ODBC (Open Database Connectivity) equivalent for Java. JDBC is a low-level interface, which means that it is used to call SQL commands directly. It is also designed to be used as a base for higher level interfaces and tools.

Your Java applet or application can talk directly to the database by using the JDBC driver to communicate with FileMaker Pro. Your SQL statements are delivered to the database and the results of those statements are sent back to you. The database can be located on another machine (the server machine) connected to the network, while your Java applet or application is located on your machine (the client machine). This is referred to as a client/server configuration.



Using the FileMaker JDBC Driver

You can use the FileMaker JDBC Driver with any Java compiler or RAD tool to connect with your database while you build the code for your Java application or applet. After the Java application or applet has been created, the FileMaker JDBC Driver must be present with the files or included within the code in order for the application or applet to communicate with the database.

To use the FileMaker JDBC Driver, your Java application or applet must register the driver with the JDBC driver manager and you must specify the correct JDBC URL from within the application or applet.You need the JDBC URL to make the connection to the database.

About the FileMaker JDBC Driver

The FileMaker JDBC Driver is a JDBC 1.2 API compatible driver designed to work with the Java Development Kit (JDK) 1.1.8. It is a Type 4 driver — a native protocol, pure Java driver that converts JDBC calls directly into the network protocol used by the database management system. This type of driver offers all the advantages of Java including automatic installation (for example, downloading the JDBC driver with an applet that uses it). The driver will work with JDK 1.1.8 and Java 2 as long as you only use JDBC 1.2 calls in a Java 2 environment.

Note Although the driver implements the entire JDBC 1.2 API, it cannot be classified as a true JDBC-compliant driver because it supports only a subset of SQL that matches the capabilities of FileMaker Pro, and is therefore not fully SQL-92 Entry Level compliant.

The FileMaker JDBC Driver is packaged as a Java archive file (with the .jar filename extension) containing a collection of class files. The archive file is named Fmpjdbc12.jar. The path to the file is:

FMWSC and Tools > FileMaker JDBC Driver > Fmpjdbc12.jar

The driver class and main entry point for the driver is named:

```
com.fmi.jdbc.JdbcDriver
```

Using a JDBC URL to connect to your database

In Java, most resources are accessed through URLs (Uniform Resource Locators). A JDBC URL is used to identify the database so the FileMaker JDBC Driver can recognize and establish a connection with the database.

The JDBC URL consists of three main parts separated by colons:

```
jdbc:<subprotocol>:<subname>
```

The first part in the JDBC URL is always the JDBC protocol ("jdbc"). The *subprotocol* is the driver name or the name of the mechanism that supports multiple drivers. In this case, the subprotocol is fmpro, which is registered with Sun Microsystems, Inc. The *subname* is the IP address of the machine that is hosting FileMaker Pro.

The FileMaker JDBC Driver connects to FileMaker Pro through an HTTP connection. The subname in the JDBC URL includes an HTTP protocol (such as HTTP or HTTPS), an IP address or domain name, and an optional port number preceded by a colon.

```
jdbc:fmpro:http://1.184.21.234:80
```

The Web Server Connector allows you to encrypt and decrypt your data via HTTPS.

jdbc:fmpro:https://www.filemaker.com:80

Here is an example of registering the FileMaker JDBC Driver with the JDBC driver manager and connecting to FileMaker Pro via the Web Companion —where the open FileMaker Pro database is named Employees.fp5 and the JDBC URL is jdbc:fmpro:http://localhost:

```
import java.sql.*;
class FMPJDBCTest
{
        public static void main(String[ ] args)
        {
            try
            {
                // register the FMPJDBC driver
                Class.forName("com.fmi.jdbc.JdbcDriver");
                // establish a connection to FileMaker Pro at a given IP
                // address
                Connection fmpConnection =
                        DriverManager.getConnection("jdbc:fmpro:
                        http://localhost", "some_user", "some_password");
                Statement fmpStatement =
                        fmpConnection.createStatement();
                // return a maximum of 50 records
                fmpStatement.setMaxRows(50);
                ResultSet resultSet = fmpStatement.execute("select \"last
                        name\", \"first name\" from \"employees.fp5\" where
                        department='engineering' order by \"last name\"");
                System.out.println("Engineering Department");
                while (resultSet.next())
                        System.out.println(resultSet.getString("last name") +
                        ", " + resultSet.getString("first name"));
            }
            catch(ClassNotFoundException classNotFoundException)
            {
                System.out.println(“Could not load driver”);
            }
            catch(SQLException sqlException)
            {
                System.out.println(“JDBC Error:” +
                        sqlException.getMessage());
            }
        }
}
```

Note This example is not meant to be compiled.

Specifying driver properties in the URL subname

You can specify the escape, fetchsize, user, and password driver properties in the subname of the JDBC URL. This is useful when you’re using a RAD tool that doesn’t support spaces, periods, or other non-letter characters.

```
jdbc:fmpro:http://17.184.21.234/properties?escape=%20.&
fetchsize=100&user=fred&password=test
```

Note These are the same properties that could be passed to the connection when calling the DriverManager.getConnection method via the Properties parameter.

Property	Description
escape	A string containing the characters to be escaped in table name, field name, and layout name SQL identifiers. The driver will escape all identifiers returned via any method in the DatabaseMetaData class. This will allow RAD tools that don't support spaces and periods in SQL identifiers to work with any FileMaker Pro database. The driver will automatically escape all identifiers for you. See “Using a character escape” on page 6-7 for more information.
fetchsize	This property allows you to set the number of records that are retrieved by the driver at any one given time. This is important for result sets (such as a result set of 20000 records) that are too large to retrieve all at once without causing memory constraints and performance problems.
user	The user name for the connection
password	The password for the connection

SQL supported by the FileMaker JDBC Driver

The FileMaker JDBC Driver provides support for certain SQL statements, a RecordID pseudo column, a ModID pseudo column, DbOpen and DbClose pseudo procedures, character escaping, and FileMaker data type mapping to JDBC SQL and Java data types.

The following is a list of the SQL statements and definitions that are supported by the FileMaker JDBC Driver.

SQL statement	Definition
SELECT statement	SELECT { { * | field_name .,.. } [, RECORDID [, MODID]] } FROM database_name [LAYOUT layout_name] [WHERE { predicate [{ { AND | OR } predicate } ...] }] [ORDER BY { field_name [ASC | DESC] } .,..] Where *predicate* equals { field_name { = | <> | > | >= | < | <= | LIKE } { value | ? } } | {field_name IS NULL} | {RECORDID = {value | ?}}

INSERT statement	INSERT INTO database_name [LAYOUT layout_name] (field_name .,..)
	VALUES ({ value | NULL | ? } .,..)

UPDATE statement	UPDATE database_name [LAYOUT layout_name]
	SET { field_name = { value | NULL | ? } } .,..
	[WHERE { predicate [{ { AND | OR } predicate } ...] }]
	Where *predicate* equals
	{ field_name { = | <> | > | >= | < | <= | LIKE } { value | ? } } | {field_name IS NULL} | {RECORDID = {value | ?} [AND MODID = { value | ? }] }

DELETE statement	DELETE FROM database_name
	[WHERE { predicate [{ { AND | OR } predicate } ...] }]
	Where *predicate* equals
	{ field_name { = | <> | > | >= | < | <= | LIKE } { value | ? } } | {field_name IS NULL} | {RECORDID = {value | ?} }

CALL stored procedure (a script) statement	{ CALL script_name (database_name [, { layout_name | password }]) } Where the outermost curly brackets { } are part of the CALL statement syntax.

Note Items within square brackets [ ] are optional and a vertical bar | means “or.” An ellipsis (…) indicates that the preceding part of the statement can be repeated any number of times. Periods and a comma (.,..) indicate that the preceding part of the statement can be repeated any number of times with the individual occurrences separated by commas. The final occurrence should not be followed by a comma.

To update a specific repeating field or field in a portal, add a period and the number of the row to the end of the field name and enclose the field name in double quotation marks. For example, to update the third repetition of the Telephone field for a record in the Employees.fp5 database, specify the following:

```
UPDATE “Employees.fp5” SET “Telephone.3”=’(555) 555-5555’ WHERE recordid=4
```

To add a specific repeating field or field in a portal, add a period and the number zero (0) to the end of the field name and enclose the field name in double quotation marks. For example, to add the City field to a portal in the Address relationship:

```
INSERT INTO “Employees.fp5” LAYOUT “Data Entry” (“Last Name”,
“Address::City.0”) VALUES (‘Jones’, ‘San Jose’)
```

Using DbOpen and DbClose pseudo procedures

The FileMaker JDBC Driver lets you open and close password-protected FileMaker Pro databases that have remote administration privileges and are located in the Web folder.

You only need to establish one connection to open or close your databases. Use “Admin” as the user name and the password that you specified in the Web Companion Configuration dialog box for remote administration. Once the databases are open, you’ll need one connection per unique database password to access the data.

Tip In order to minimize the number of connections, assign the same database password for all your databases.

Here is an example for opening and closing a password-protected database named “inventory.fp5,” where the remote administration user name is “Admin,” the remote administration password (set in the Web Companion Configuration dialog box) is “admin,” and the database password is “inventory.”

```
import java.sql.*;

/**

public class FMPJDBCSecurity

{

        public static void main(String[ ] args)

        {
            try
            {
                // register the FMPJDBC driver

                Class.forName("com.fmi.jdbc.JdbcDriver");

                // establish a connection to FileMaker Pro for remote
                // administration purposes. The user name for remote
                // administration is always "Admin."

                Connection adminConnection =
                              DriverManager.getConnection("jdbc:fmpro:
                              http://localhost", "Admin", "admin");

                // create a statement for opening the "inventory.fp5"
                // database. Since the "inventory.fp5" database is
                // password protected, you also need to specify the
                // password in the call to the dbopen procedure.

                CallableStatement openDbStatement =
                              adminConnection.prepareCall("{call
                              dbopen(\"inventory.fp5\", \"inventory\")}");

                openDbStatement.execute();

                // establish a connection to FileMaker Pro for
                // retrieving data from the "inventory.fp5" database.
                // The "inventory.fp5" database uses FileMaker Pro
                // security, so you don’t need to provide a user name.
                // A user name is only necessary for remote
                // administration or if the databases is protected via
                // the Web Security Database.

                Connection inventoryConnection =
                              DriverManager.getConnection("jdbc:fmpro:
                              http://localhost", null, "inventory");

                Statement inventoryStatement =
                              inventoryConnection.createStatement();

                ResultSet inventoryData =
                              inventoryStatement.executeQuery("select *
                              from \"inventory.fp5\"");

                // create a statement for closing the "inventory.fp5"
                // database

                CallableStatement closeDbStatement =
                        adminConnection.prepareCall("{call
                        dbclose(\"inventory.fp5\")}");

                closeDbStatement.execute();

            }

            catch(ClassNotFoundException classNotFoundException)
            {
                System.out.println("Could not load driver");
            }
            catch(SQLException sqlException)
            {
                System.out.println("JDBC Error: " +
                        sqlException.getMessage());
            }
        }
}
```

Using the RecordID pseudo column

The FileMaker JDBC Driver provides a RecordID pseudo column (in place of a primary key used by other types of databases) that can be specified in the column name list of a SELECT statement or in the WHERE clause of SELECT, UPDATE or DELETE statements. This lets you guarantee that the statement will operate on a specific record.

All other columns are ignored when the RecordID pseudo column is used in a WHERE clause.

```
UPDATE "Employees.fp5" SET department='engineering' WHERE
recordid=4
```

Using the ModID pseudo column

Each record in a FileMaker Pro database has a corresponding modification ID (ModID) number that increases incrementally every time the record is modified. To detect modification collisions, the FileMaker JDBC Driver provides a ModID pseudo column that can be used in the WHERE clause of an UPDATE statement in conjunction with the RecordID. The Web Companion compares the ModID in the WHERE clause to the current ModID of the record and an error is returned if they do not match.

```
...
Connection connection = DriverManager.getConnection("jdbc:fmpro:
        http://localhost", "some_user", "some_password");
Statement statement = connection.createStatement();
// retrieve all of the records where department equals "engineering"
ResultSet resultSet = statement.executeQuery("SELECT recordid,
        modid, "last name", "first name" FROM \"Employees.fp5\" WHERE
        department='engineering'");
// create an UPDATE statement for changing the department to "software
// engineering"
PreparedStatement preparedStatement =
        connection.prepareStatement("UPDATE \"Employees.fp5\" SET
        department='software engineering' WHERE recordid=? AND
        modid=?");
while (resultSet.next())
{
        // set the recordid parameter
        preparedStatement.setString(1,
                resultSet.getString("RECORDID"));
        // set the modid parameter
        preparedStatement.setString(2, resultSet.getString("MODID"));
        // change the department from "engineering" to "software
        // engineering"
        preparedStatement.executeUpdate();
}
...
```

SQL statement examples

The following are some examples of SQL statements, some of which use RecordID and ModID pseudo columns, and DbOpen and DbClose pseudo procedures:

```
SELECT recordid, modid, "last name", "first name", department FROM
"Employees.fp5" WHERE "last name"='smith' AND "first name" = 'joe'

SELECT * FROM "Employees.fp5" WHERE recordid=4

SELECT recordid, modid, * FROM "employees.fp5"
SELECT "last name", "first name", "telephone::phone number" FROM
"employees.fp5" LAYOUT "personal info"
```

```
UPDATE "Employees.fp5" SET department='engineering' WHERE
recordid=4 AND modid=2
```

```
UPDATE "Employees.fp5" LAYOUT "personal info" SET
"telephone::phone number.2"='555-555-5555' WHERE recordid=4
```

```
DELETE FROM "Employees.fp5" WHERE recordid=4
```

```
{ CALL DbOpen("inventory.fp5", "some password") }
```

```
{ CALL DbClose("inventory.fp5") }
```

```
{ CALL FindManagers("employees.fp5") }
```

```
{ CALL SortByLastName("employees.fp5", "list view") }
```

Using a character escape

The FileMaker JDBC Driver supports escaping of lower ASCII characters in column and table name SQL identifiers. This is useful if your RAD tool doesn’t support characters such as spaces in column names or periods in table names. The escape sequence starts with the dollar symbol ($) and is followed by the two-digit hex value for the character (such as 2E for a period and 20 for a space).

```
employees.fp5 => employees$2Efp5
```

```
last name => last$20name
```

```
SELECT last$20name FROM employees$2Efp5
```

FileMaker data type mapping to JDBC SQL and Java data types

The FileMaker JDBC Driver uses the following mappings when converting FileMaker Pro data types to JDBC SQL types or to Java data types. (For information about these types, see the JDK 1.1.8 documentation web pages at www.javasoft.com.)

This FileMaker Pro data type	Converts to this JDBC SQL type
TEXT	java.sql.Types.LONGVARCHAR
NUMBER	java.sql.Types.DOUBLE
DATE	java.sql.Types.DATE
TIME	java.sql.Types.TIME
CONTAINER	java.sql.Types.LONGVARBINARY

This FileMaker Pro data type	Converts to this Java data type
TEXT	java.lang.String
NUMBER	java.lang.Double
DATE	java.sql.Date
TIME	java.sql.Time
CONTAINER	java.awt.Image
Repeating and related fields	com.fmi.jdbc.Array

FileMaker Pro support for Unicode characters

FileMaker Pro only supports the Windows Latin 1 and Macintosh character sets, which are a subset of Unicode. Therefore, any character data submitted to FileMaker Pro that contains characters not present in these character sets (such as certain math symbols) will not be stored properly in your database. FileMaker Pro inserts a question mark (?) for any character that it does not recognize.

About the FileMaker JDBC Driver interfaces and extensions

The FileMaker JDBC Driver implements all of the following JDBC interfaces:

- CallableStatement
- Connection
- DatabaseMetaData
- Driver

- PreparedStatement
- ResultSet
- ResultSetMetaData
- Statement

The following FileMaker Pro-specific extensions have been added:

This JDBC interface	Includes this FileMaker Pro extension
java.sql.DatabaseMetaData	com.fmi.jdbc.DatabaseMetaDataExt
java.sql.ResultSetMetaData	com.fmi.jdbc.ResultSetMetaDataExt

The following classes have been added in support of the FileMaker Pro extensions:

Class name	Description
com.fmi.fmpdb.FMPError	FileMaker Pro error codes
com.fmi.fmpdb.FMPLayoutField	Information associated with a field on a layout
com.fmi.fmpdb.FMPLayoutFieldEnumerator	Class for enumerating the fields on a layout
com.fmi.fmpdb.FMPLayoutMetaData	Metadata for a given layout
com.fmi.jdbc.Array	Class used to represent repeating and related fields

The API documentation for these standard interfaces and the FileMaker extensions is included in HTML format on the *FMWSC and Tools* CD:

FMWSC and Tools > JDBC and Java > JDBC > FileMaker JDBC Driver > Documentation

Example 1: Looking at the FileMaker Pro Explorer application

This developer-tool-independent example is a Java application used for displaying FileMaker Pro database information, similar to the Windows Explorer and Mac OS Finder applications. You can use the FileMaker Pro Explorer application along with the FileMaker JDBC Driver to view any open database on any computer that's shared via the Web Companion, by specifying the JDBC URL that includes the IP address of the computer where FileMaker Pro is running. You can view the application's source code in any text editor or Java editing tool.

The application was created using the basic Java classes to display a database tree, and FileMaker Pro-specific extensions have been added to provide detailed information about the fields and layouts. The user interface was created using the Swing 1.1.1 class library—an add-on to the Java Development Kit (JDK) 1.1.8.

For information on the Swing class library, go to the Sun Microsystems web site at www.javasoft.com.

Setup requirements

Included with the example is the swingall.jar file, on the *FMWSC and Tools* CD.

FMWSC and Tools > JDBC and Java > JDBC > FileMaker JDBC Driver > Examples > FileMaker Explorer > swingall.jar

To view the example on Windows machines, you need:

- Java.exe (included with JDK 1.1.8 for Windows) or equivalent Java virtual machine installed in the system path on your computer

To view the example on Mac OS machines, you need:

- MRJ 2.2 or equivalent Java virtual machine installed on your computer

Install the example and the FileMaker JDBC Driver

If necessary, install the FileMaker Explorer example and the FileMaker JDBC Driver.

FMWSC and Tools > FileMaker JDBC Driver > JDBC Examples > FileMaker Explorer

FMWSC and Tools > FileMaker JDBC Driver > Fmpjdbc12.jar

Open and share your databases via the Web

1. In FileMaker Pro, open any FileMaker Pro database, such as the Inventory.fp5 database located in the JBuilder example folder:

FMWSC and Tools > FileMaker JDBC Driver > JDBC Examples > JBuilder 3.0 Professional > Inventory.fp5

2. Choose File menu > Sharing, verify that Web Companion is selected, and click OK to share the database on the Web.

For information about setting up the Web Companion so that it's already selected in this dialog box, see "Enabling the Web Companion" on page 3-3.

Run the FileMaker Pro Explorer application

The FileMaker Pro Explorer application is located in the FileMaker Explorer folder:

FMWSC and Tools > FileMaker JDBC Driver > Examples > FileMaker Explorer > FileMakerExplorer.bat

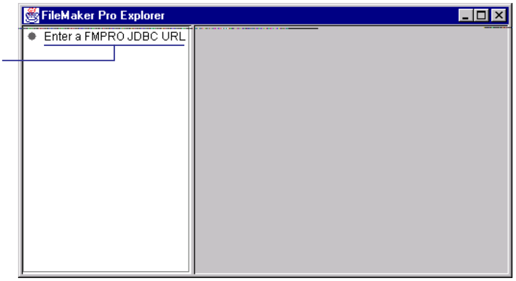
FMWSC and Tools > FileMaker JDBC Driver > Examples > FileMaker Explorer > FileMakerExplorer

1. Start the FileMaker Pro Explorer application by doing one of the following:

- Windows: Choose Start menu > Run and locate and select the FileMakerExplorer.bat file. Then, add the location of the JDK for running the application at the end of the command line:

```
"full path\FileMakerExplorer.bat" c:\jdk1.1.8
```

- Mac OS X: Double-click the application icon to start FileMaker Pro Explorer.



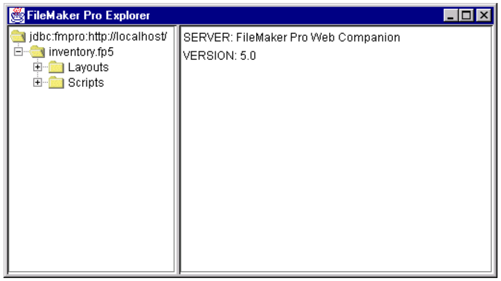
Replace entire text with the JDBC URL for the computer running the database

2. Click the root node of the tree in the left side of the FileMaker Pro Explorer window to select it, and replace the entire text with the JDBC URL for the computer that is running the database, for example, type `jdbc:fmpro:http://localhost/`. Then press Enter.

See "Using a JDBC URL to connect to your database" on page 6-2 for information.

3. Select Inventory.fp5 (or other database name) on the left side of the window to see information about it on the right side.

4. Expand the Inventory.fp5 node (folder) to expose the Layouts and Scripts nodes.

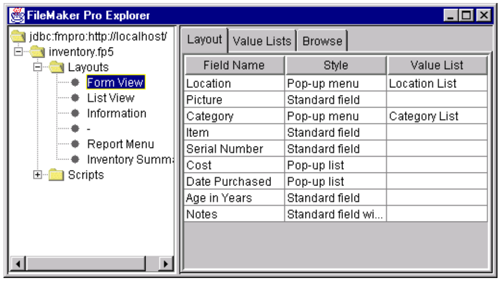


```
SERVER: FileMaker Pro Web Companion
VERSION: 5.0
```

5. Select Layouts to display the names of the layouts for the Inventory.fp5 database on the right side of the window.

6. Expand the Layouts node to display a leaf node for each layout.

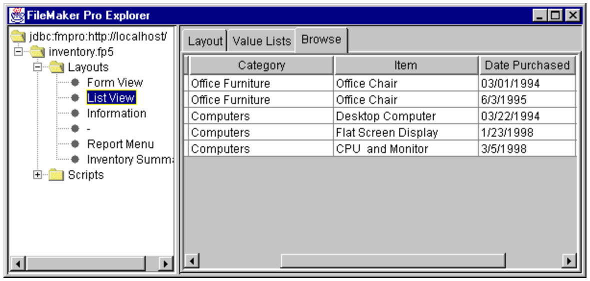
7. Select the Form View node to display the fields for that layout on the right side of the window.



Field Name	Style	Value List
Location	Pop-up menu	Location List
Picture	Standard field	
Category	Pop-up menu	Category List
Item	Standard field	
Serial Number	Standard field	
Cost	Pop-up list	
Date Purchased	Pop-up list	
Age in Years	Standard field	
Notes	Standard field wi...	

8. Click the Browse tab to display the first five records in the Inventory.fp5 database.

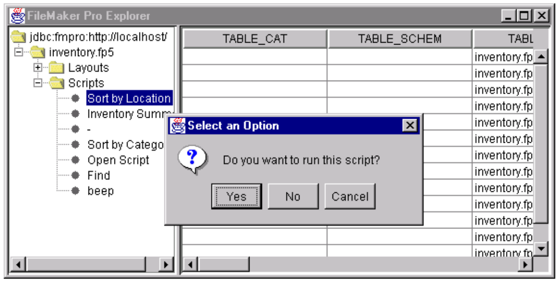
The columns correspond to the fields available for the selected layout.



Category	Item	Date Purchased
Office Furniture	Office Chair	03/01/1994
Office Furniture	Office Chair	6/3/1995
Computers	Desktop Computer	03/22/1994
Computers	Flat Screen Display	1/23/1998
Computers	CPU and Monitor	3/5/1998

9. Select the Scripts node to display the names of scripts for the Inventory.fp5 database on the right side of the window.

When you select a script name, a warning dialog box appears. This gives you the opportunity to not run the script, which is especially important if a database includes a script for deleting or modifying records.

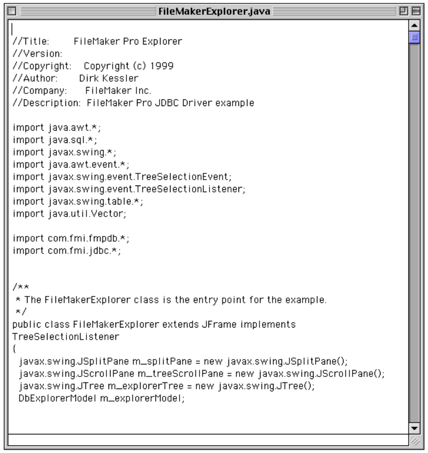


Do you want to run this script?

View the source code of the example

1. Start your text editor (such as Notepad or SimpleText) or Java development tool.

2. Open the main class file, FileMakerExplorer.java, in the FileMaker Explorer folder.



```
//Title:       FileMaker Pro Explorer
//Version:
//Copyright:   Copyright (c) 1999
//Author:      Dirk Kessler
//Company:     FileMaker Inc.
//Description:  FileMaker Pro JDBC Driver example

import java.awt.*;
import java.sql.*;
import javax.swing.*;
import java.awt.event.*;
import javax.swing.event.TreeSelectionEvent;
import javax.swing.event.TreeSelectionListener;
import javax.swing.table.*;
import java.util.Vector;

import com.fmi.fmpdb.*;
import com.fmi.jdbc.*;


/**
 * The FileMakerExplorer class is the entry point for the example.
 */
public class FileMakerExplorer extends JFrame implements
TreeSelectionListener
{
  javax.swing.JSplitPane m_splitPane = new javax.swing.JSplitPane();
  javax.swing.JScrollPane m_treeScrollPane = new javax.swing.JScrollPane();
  javax.swing.JTree m_explorerTree = new javax.swing.JTree();
  DbExplorerModel m_explorerModel;
```

The source code is well commented, describing the methods for each class used in this example.

Example 2: Creating the JBuilder Inventory application

This example demonstrates how to build a Java front end to a FileMaker database using the development tool, JBuilder 3.0 Professional for Windows and the FileMaker JDBC Driver. This example uses a modified version of the Asset Management.fp5 database that ships with FileMaker Pro. The following steps are for creating a Java application that accesses the database, renamed Inventory.fp5.

Install the example and FileMaker JDBC Driver

If necessary, install the JBuilder folder of example files and the FileMaker JDBC Driver.

FMWSC and Tools > FileMaker JDBC Driver > JDBC Examples > JBuilder 3.0 Professional

FMWSC and Tools > FileMaker JDBC Driver > Fmpjdbc12.jar

The JBuilder 3.0 Professional example folder contains the database file used in this example and all of the completed files generated by the JBuilder wizard for the application.

Set up JBuilder to use the FileMaker JDBC Driver

1. In a text editor (such as Notepad), open the Jbuilder.ini file from the bin folder inside the JBuilder 3.0 Professional application folder. (Please make a backup copy of this file before proceeding with these instructions.)

2. In the [Java_VM_Properties] section, add the path of the Fmpjdbc12.jar file to the end of the Djava.class.path line.

For instructions on installing the FileMaker JDBC driver into the JBuilder environment, see “Installing and setting up JBuilder for database applications” in the online JBuilder Help or go to www.borland.com/devsupport/jbuilder/.

Open and share the Inventory.fp5 database

1. In FileMaker Pro, open the Inventory.fp5 file in the JBuilder 3.0 Professional folder:

FMWSC and Tools > FileMaker JDBC Driver > JDBC Examples > JBuilder 3.0 Professional > Inventory.fp5

2. Choose File menu > Sharing, verify that Web Companion is selected, and click OK to share the database on the Web.

For information about setting up the Web Companion so that it’s already selected in this dialog box, see “Enabling the Web Companion” on page 3-3.

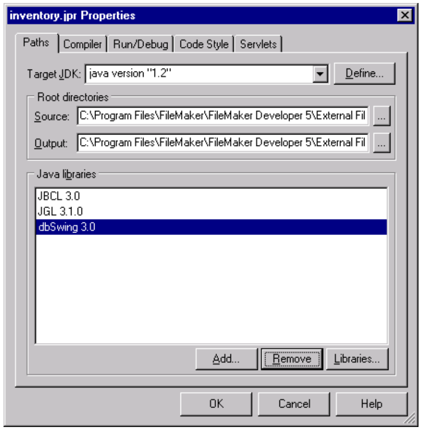
Start a new JBuilder project

1. Start JBuilder 3.0 Professional for Windows.

2. In JBuilder, choose File menu > New Project.

3. In the dialog box, specify the name and location for the project file (JBuilder Inventory).

4. Choose Project menu > Properties.



inventory.jpr Properties

Target JDK: java version "1.2"

Root directories

Source: C:\Program Files\FileMaker\FileMaker Developer 5\External Fil

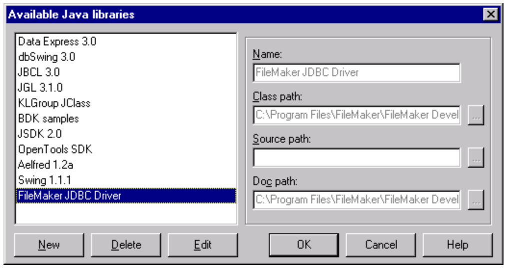
Output: C:\Program Files\FileMaker\FileMaker Developer 5\External Fil

Java libraries

JBCL 3.0
JGL 3.1.0
dbSwing 3.0

5. Click Libraries in the Properties dialog box.

6. In the Available Java Libraries dialog box, select the Fmpjdbc12.jar file. (If the FileMaker JDBC Driver does not appear in this dialog box, click New to locate the file and add it to the list.) Then click OK.



Name: FileMaker JDBC Driver

Class path: C:\Program Files\FileMaker\FileMaker Devel

Source path:

Doc path: C:\Program Files\FileMaker\FileMaker Devel

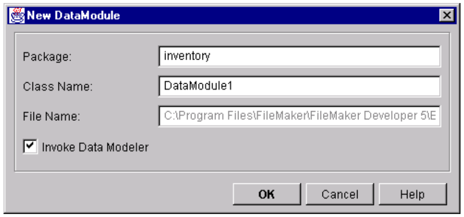
7. Click OK to close the Properties dialog box.

Create the data module

1. In JBuilder, choose File menu > New.

2. In the New dialog box, select the Data Module icon and click OK.

3. In the New Data Module dialog box, make sure that inventory is in the Package text box and that the checkbox for Invoke Data Modeler is selected.



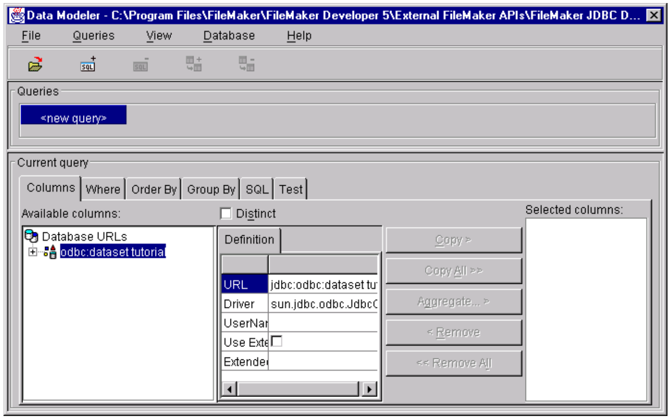
Package: inventory

Class Name: DataModule1

File Name: C:\Program Files\FileMaker\FileMaker Developer 5\E

☑ Invoke Data Modeler

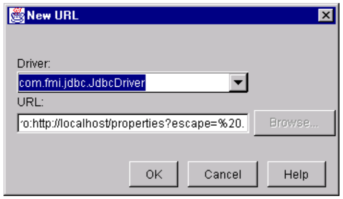
4. Click OK to open the Data Modeler.



URL	jdbc:odbc:dataset tu
Driver	sun.jdbc.odbc.JdbcO
UserNar	
Use Exte	□
Extende	

Design the data module

1. In the Data Modeler, choose Database menu > Add Connection URL.



Driver: com.fmi.jdbc.JdbcDriver

URL: o:http://localhost/properties?escape=%20.

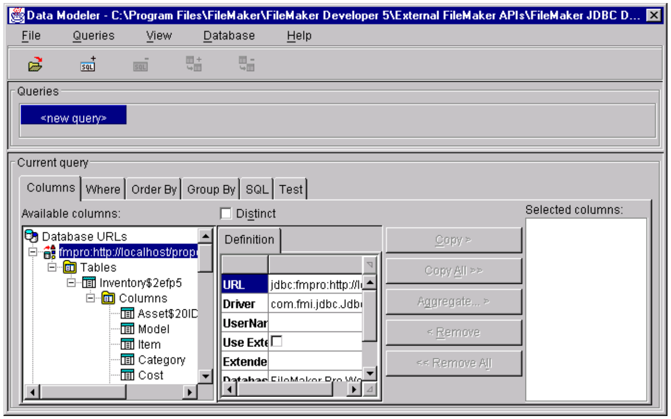
2. In the Driver text box, type `com.fmi.jdbc.JdbcDriver`.

3. In the URL text box, type `jdbc:fmpro:http://localhost/properties?escape=.%20`.

Because JBuilder's Application Generator doesn't allow for periods in table names or spaces in column names for certain operations, you need to add escape properties for them in the subname of the URL.

4. Click OK to close the Add Connection URL dialog box.

5. In the Data Modeler, click the + symbol to expand the URL node that you added.



URL	jdbc:fmpro:http://l
Driver	com.fmi.jdbc.Jdb
UserNar	
Use Exte	□
Extende	
Databas	FileMaker Pro Wi

6. In the User Authentication dialog box, leave the User Name and Password text boxes blank (the database is not password protected) and click OK.

JBuilder connects to FileMaker Pro.

7. In the Data Modeler, click the + symbol to expand Tables.

8. Click the + symbol to expand the Inventory$2efp5 table (now encoded with the period escaped).

9. Click the + symbol to expand Columns.

10. Select each of the following columns and copy it to the right side of the Data Modeler.

Item
Category
Location
Cost
Date Purchased
Picture
Serial Number
Information

Test the data module

1. In the Data Modeler, click the Test tab and click Execute Query.

The data and all of the images from the Inventory.fp5 database are downloaded into the data module.

2. Choose File menu > Save.

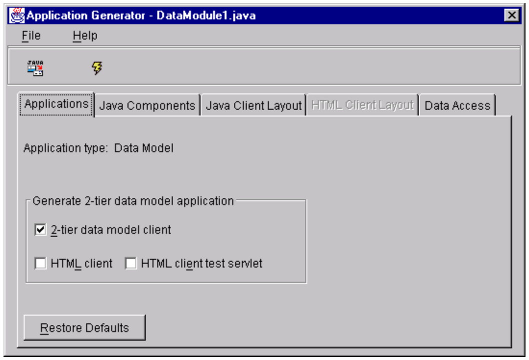
JBuilder saves the data module and automatically adds it to your project.

3. Choose File menu > Exit.

4. Click Yes to invoke the Application Generator.

Generate the application

1. In the Application Generator, click Restore Defaults and make sure that 2-tier data model client is selected.



Application type: Data Model

Generate 2-tier data model application

☑ 2-tier data model client

□ HTML client □ HTML client test servlet

2. Click the Java Client Layout tab.

Because JBuilder requires a unique row identifier for updating and deleting rows, it automatically adds the RecordID pseudo column to the list of output columns, which you probably don't want displayed in your application.

3. Deselect the Layout check mark for the RecordID identifier.

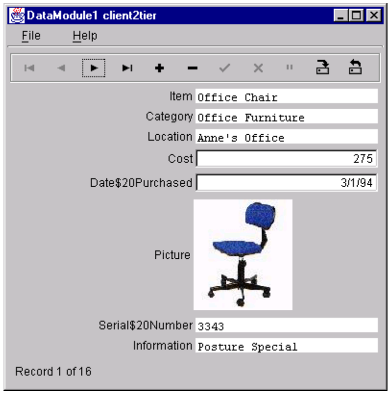
4. Choose File menu > Generate.

5. Click Generate.

The Application Generator generates the source code and adds the generated files to your project.

6. Click Close All.

7. Choose Run menu > Run Application to execute the example application.



Item Office Chair
Category Office Furniture
Location Anne's Office
Cost 275
Date$20Purchased 3/1/94
Picture
Serial$20Number 3343
Information Posture Special
Record 1 of 16

The field attributes used for the example from the Inventory.fp5 database are described in this table:

Field type	Field name	Field attribute
Text	Item	User defined entry field
Text	Category	Value List/Category List (Pop-up menu) Office Furniture Computers Telephones Vehicles Copier Printer AV
Text	Location	Value List/Location List (Pop-up menu) Fred’s Office Dirk’s Office Pedro’s Office Anne’s Office Julie’s Office Ruth’s Office Joanna’s Office Business Center
Number	Cost	User defined entry field
Date	Date Purchased	User defined entry field
Container	Picture	Graphic import
Text	Serial Number	User defined entry field
Text	Information	User defined entry field

Example 3: Creating the Visual Cafe Inventory application

This example demonstrates how to build a Java front end to a FileMaker database using the development tool Symantec’s Visual Cafe 4.0 Expert Edition for Windows and the FileMaker JDBC Driver. The following steps are for creating a Java application that accesses the inventory_db database.

Note Because the Visual Cafe DataBound Project Wizard does not support spaces in database filenames or field names, or periods and spaces in table names, this example uses a modified version of the Asset Management.fp5 database that ships with FileMaker Pro, called inventory_db (with no filename extension).

Install the example and the FileMaker JDBC Driver

If necessary, install the Visual Cafe example and the FileMaker JDBC Driver.

FMWSC and Tools > FileMaker JDBC Driver > JDBC Examples > Visual Cafe 4.0 Expert Edition

FMWSC and Tools > FileMaker JDBC Driver > Fmpjdbc12.jar

The Visual Cafe folder contains the modified database file used in this example and all of the completed files generated by the DataBound Project Wizard for the application.

Set up Visual Cafe to use the FileMaker JDBC Driver

1. Start Visual Cafe 4.0 Expert Edition for Windows.

2. Choose Tools menu > Environment Options and click the Internal VM tab. Then click New for the Classpath Settings.

3. Locate the Fmpjdbc12.jar file and add it to the SC.INI CLASSPATH list to add the FileMaker JDBC Driver to the Visual Cafe environment.

Open and share the inventory_db database

1. In FileMaker Pro, open the inventory_db file in the Visual Cafe example folder:

FMWSC and Tools > FileMaker JDBC Driver > JDBC Examples > Visual Cafe 4.0 Expert Edition > inventory_db

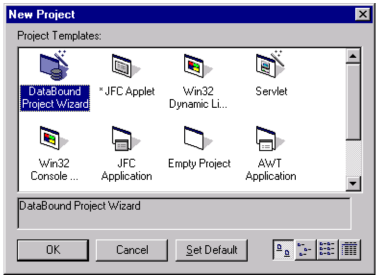
2. Choose File menu > Sharing, verify that Web Companion is selected, and click OK to share the database on the Web.

For information about setting up the Web Companion so that it's already selected in this dialog box, see "Enabling the Web Companion" on page 3-3.

Create a new Visual Cafe project

1. In Visual Cafe, choose File menu > New Project.

2. Select the DataBound Project Wizard template and click OK.



DataBound Project Wizard

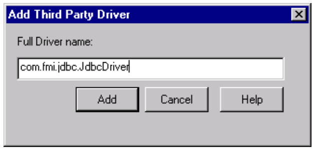
3. In the DataBound Project Wizard, select Application for the project type and click Next.

4. Select Define another data source and click Next.

5. In the DataBound Project Wizard, click New.

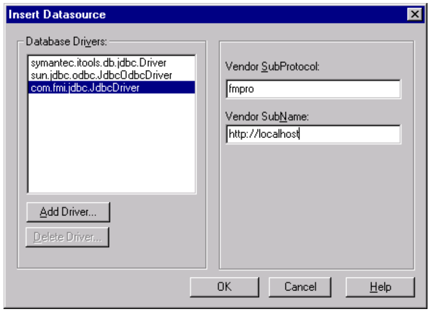
6. In the Insert Datasource dialog box, click Add Driver.

7. In the Add Third Party Driver dialog box, type `com.fmi.jdbc.JdbcDriver` and click Add.



Full Driver name: com.fmi.jdbc.JdbcDriver

8. In the Insert Datasource dialog box, select com.fmi.jdbc.JdbcDriver, type `fmpro` in the Vendor SubProtocol box, type `http://localhost` in the Vendor SubName box (or the IP address of the computer hosting the inventory_db database), and then click Add Driver.



Database Drivers:
- symantec.itools.db.jdbc.Driver
- sun.jdbc.odbc.JdbcOdbcDriver
- com.fmi.jdbc.JdbcDriver

Vendor SubProtocol: fmpro

Vendor SubName: http://localhost

9. In the DataBound Project Wizard, select com.fmi.jdbc.JdbcDriver in the list of data sources and click Next.

10. In the User Authentication dialog box, click OK.

The inventory_db database is not password protected.

11. In the DataBound Project Wizard, select inventory_db in the list of database tables and views, and click Next.

12. Click Clear to move all the used columns to the Available Columns list.

13. Select the following columns and move them to the Used Columns list by selecting each one and clicking Move.

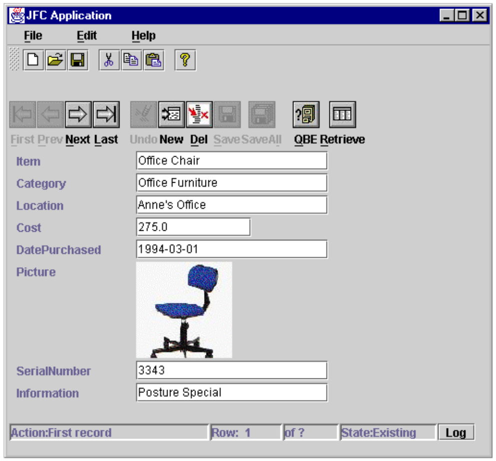
Item
Category
Location
Cost
Date Purchased
Picture
Serial Number
Information

14. Click Next.

15. In the DataBound Project Wizard, select Picture and choose ImageViewer from the Component list to change the UI component used for displaying the Picture column.

16. Click Finish to generate the code.

17. Choose Project menu > Execute to run the application.



Item: Office Chair
Category: Office Furniture
Location: Anne's Office
Cost: 275.0
DatePurchased: 1994-03-01
Picture
SerialNumber: 3343
Information: Posture Special

Action:First record | Row: 1 | of ? | State:Existing

The field attributes used for the example from the inventory_db database are described in the following table:

Field type	Field name	Field attribute
Text	Item	User defined entry field
Text	Category	Value List/Category List (Pop-up menu) Office Furniture Computers Telephones Vehicles Copier Printer AV

Field type	Field name	Field attribute
Text	Location	Value List/Location List (Pop-up menu) Fred's Office Dirk's Office Pedro's Office Anne's Office Julie's Office Ruth's Office Joanna's Office Business Center
Number	Cost	User defined entry field
Date	Date Purchased	User defined entry field
Container	Picture	Graphic import
Text	Serial Number	User defined entry field
Text	Information	User defined entry field

Using the FileMaker Java classes

The FileMaker Pro 4.0 Java classes are designed to retrieve FileMaker data from your database via the FileMaker Java API. They do not support SQL and they're not supported by any RAD tool. However, if you used these proprietary FileMaker Java classes from the previous versions of FileMaker Developer 5 or the FileMaker Pro Developer Edition to create your Java front-ends to your database, you may wish to continue using them for updating your Java applications or applets.

Important The Web Companion does not support non-number characters in number fields when it converts FileMaker Pro data into the Java class format. Any Java applications or applets that you created using the FileMaker Pro 4.0 Java classes will lose all characters other than numbers (such as the dollar sign $) in number fields when they access a FileMaker Pro database.

The FileMaker Java Class Library is included on the *FMWSC and Tools* CD.

FMWSC and Tools > FileMaker JDBC Driver > Java Class Library

About the FileMaker Java Class Library

The FileMaker Java Class Library is a set of Java classes that sends requests to the FileMaker Pro Web Companion and gives you access to the results.

FileMaker Java classes communicate with the Web Companion using standard HTTP (Hypertext Transfer Protocol). The classes formulate a request string based on the parameters you specify. Then the classes send the request to the Web Companion. The response is generated and sent back in the CDML proprietary format and processed by the FMProResponse class. The Java application or applet then determines how to display the data.

There are three main Java classes in the library:

- FMProRequest
- FMProProxy
- FMProResponse

Use the FMProRequest class to create objects that submit queries to the database. You specify the name of the database, layout, and fields in a FMProRequest object.

Use the FMProProxy class to create objects that execute the queries. The FMProProxy object is used to send a query from an FMProRequest object to the computer where FileMaker Pro is hosting the database, and to receive the result of the query from the Web Companion. The FMProProxy object then converts the resulting data into an FMProResponse object.

You can use a FMProProxy object to retrieve the following:

- the names of all open databases shared via the Web Companion

- the names of layouts, fields in a layout, and scripts defined for a database
- the images from FileMaker Pro container fields

The FMProResponse class allows you to access the query results. The information you have access to includes:

- a result code indicating if the query was successfully processed
- the number of records returned
- the total number of records in the database
- the names, field types, and value lists (if applicable) for each of the fields on the specified layout, and field data
- record IDs for the returned records

This information can be retrieved using the various methods belonging to the FMProResponse object.

Looking at the Java applet examples

There are two examples of Java applets that use the FileMaker Java classes to communicate with a FileMaker Pro database. The FMBanner applet is a simple Java applet that displays a textual or picture-based advertisement based on data from a FileMaker Pro database. The FMMemoPad applet allows you to browse, edit, find, sort, and create records in a FileMaker Pro database.

Note To view these examples, you need a web browser that is compatible with Java 1.1, such as Internet Explorer 4.0 or Netscape Communicator 4.0 with the JDK 1.1 support patch.

To view the FMBanner applet:

1. Locate the FMBanner folder on the *FMWSC and Tools* CD.

FMWSC and Tools > FileMaker JDBC Driver > Java Class Library > JDBC Examples > FMBanner

2. Place a copy of the FMBanner.html and FMBanner.jar files into the Web folder in the FileMaker Pro 5.5 application folder.

FileMaker Pro 5.5 > Web > FMBanner.html
FileMaker Pro 5.5 > Web > FMBanner.jar

3. In FileMaker Pro, open the FMBanner.fp5 file.

4. Choose File menu > Sharing, verify that Web Companion is selected, and click OK to share the database on the Web.

For information about setting up the Web Companion so that it's already selected in this dialog box, see "Enabling the Web Companion" on page 3-3.

5. In your web browser, type `localhost` or your computer's IP address followed by /`FMBanner.html` and press Enter.

http://localhost/FMBanner.html

For information on setting up your computer as a localhost, see "Testing your site without a network connection" on page 3-18.

Changes you make to the FMBanner.fp5 database are reflected when the Java applet loads in the browser. Images are displayed in the Picture field—if no image is in the field, then text from the String field is displayed. You can set the font type and size of the text, and change the background color for each advertisement. If you specify a URL in the URL field, clicking the advertisement will load the URL. The database contains a Temp Pict field that you can use to store images that you don't want to display in the advertisement.

To view the FMMemoPad applet:

1. Locate the FMMemoPad folder on the *FMWSC and Tools* CD.

FMWSC and Tools > FileMaker JDBC Driver > Java Class Library > JDBC Examples > FMMemoPad

2. Place a copy of the FMMemoPad.html and FMMemoPad.jar files into the Web folder in the FileMaker Pro 5.5 application folder.

FileMaker Pro 5.5 > Web > FMMemoPad.html
FileMaker Pro 5.5 > Web > FMMemoPad.jar

3. In FileMaker Pro, open the FMMemoPad.fp5 file.

4. Choose File menu > Sharing, verify that Web Companion is selected, and click OK to share the database on the Web.

For information about setting up the Web Companion so that it appears selected in this dialog box, see “Enabling the Web Companion” on page 3-3.

5. In your web browser, type `localhost` or your computer’s IP address followed by `/FMMemoPad.html` and press Enter.

http://localhost/FMMemoPad.html

For information on setting up your computer as a localhost, see “Testing your site without a network connection” on page 3-18.

6. After the applet has loaded, click Open Memo Pad to display the digital memo pad in a separate window.

The FMMemoPad applet lets you browse, edit, find, sort, and add records to the FMMemoPad.fp5 database.

This page intentionally left blank.

Appendix A
Valid names used in CGI requests for FileMaker XML data

This appendix describes the valid names of requests and their parameters you can use in a CGI (Common Gateway Interface) command when requesting FileMaker Pro data in XML format. Included are HREF link and HTML form examples of each request. For scripting examples of each request, see the JavaScript library, FMP.js, in the XML Examples folder (on the *FMWSC and Tools* CD).

The CGI command requesting FileMaker Pro data in XML format always begins with the action as FMPro? and the format file as either –dso_xml, –dso_xml_dtd, –fmp_xml, or –fmp_xml_dtd.

The following is a list of the request names and parameters:

Request names	Parameter names
–find	–db
–findall	–lay
–findany	–format
–view	–recid
–new	–modid
–edit	–lop, –op
–delete	–max
–dbnames	–skip
–layoutnames	–sortorder, –sortfield
–scriptnames	–script, –script.prefind, –script.presort
–dbopen	–styletype, –stylehref
–dbclose	–password

Note The name of a database field can also be used in a CGI command. It is not a parameter or a request name, and therefore is not preceded by a hyphen (–).

Generating a –find, –findall, or –findany request

Name/Value Type: Find Record(s) Request

What it does: Submits a search request using defined criteria.

A web user must have browsing access privileges with the database in order to execute these requests.

Required parameters: –db, –format

Optional parameters: –lay, –recid, –lop, –op, –max, –skip, –sortorder, –sortfield, –script, –script.prefind, –script.presort, field name, –styletype, –stylehref

Examples of –find, –findall, and –findany requests

To find a record using a hypertext link:

```
<a href="FMPro?–db=employees.fp5&–format=–fmp_xml
&Country=USA&–max=1&–find">Find first USA record</a>
```

To find all records in the database using a hypertext link:

```
<a href="FMPro?–db=employees.fp5&–format=–fmp_xml &
–findall">Find all records</a>
```

To find any record using a hypertext link:

```
<a href="FMPro?–db=employees.fp5&–format=–fmp_xml &
–findany">Find a random record for today’s daily quote</a>
```

To find some records using a form action:

```
<form action="FMPro" method="post">
<input type="hidden" name="–db" value="employees.fp5">
```

```
<input type="hidden" name="–format" value="–fmp_xml">
<input type="hidden" name="–max" value="all">
<input type="text" size=12 name="Country" value="USA">
<input type="submit" name="–find" value="Find Records">
</form>
```

Generating a –view request

Name/Value Type: View Request

What it does: Retrieves layout information from a database and displays it in the FMPXMLLAYOUT grammar.

Required parameters: –db, –lay, –format (= –fmp_xml)

Optional parameters: –styletype, –stylehref

Examples of –view requests

To retrieve layout information using a hypertext link:

```
<a href="FMPro?–db=employees.fp5&–lay=LayoutOne&–format= –
fmp_xml&–view">"Take me to a search page"</a>
```

To retrieve layout information using a form action:

```
<form action="FMPro" method="post">
<input type="hidden" name="–db" value="employees.fp5">
<input type="hidden" name="–lay" value="LayoutOne">
<input type="hidden" name="–format" value="–fmp_xml">
<input type="submit" name="–view" value="Show Search Page">
</form>
```

Generating a –new request

Name/Value Type: New Record Request

What it does: Creates a new record and populates that record with the contents of any field name/value pairs.

A web user must have access privileges for creating records in order to execute this request.

Required parameters: –db, –format, one or more field name(s)

Optional parameters: –styletype, –stylehref

Note To include new data for a portal, you must also specify the layout and the relationship name for the related database followed by two colons. You can only add one row/record to a portal per request. (See “Note The -max parameter now returns 0 if the request returns no records.” on page 5-9 for information.)

Examples of –new requests

To create a new record using a hypertext link:

```
<a href="FMPro?–db=employees.fp5&Country=Australia&
–format= –fmp_xml&–new">
```

To create a new record using a form action:

```
<form action="FMPro" method="post">
<input type="hidden" name="–db" value="employees.fp5">
<input type="hidden" name="–format" value=”–fmp_xml”>
<input type="text" size=12 name="Country" value="Australia">
<input type="submit" name="–new" value="New Record">
</form>
```

Generating an –edit request

Name/Value Type: Edit Record Request

What it does: Updates the record specified by –recid, populating the fields with the contents of any field name/value pairs.

The –recid parameter indicates which record should be edited. In order to edit a record, the web user must have editing privileges for the database.

Required parameters: –db, –format, –recid, one or more field name(s)

Optional parameters: –modid, –styletype, –stylehref

Note To edit records in a portal, you must also include the layout name and the relationship name followed by two colons. To specify each record in the portal, include a period and a consecutive number after the related field name, such as address::city.1 for the first row (record) in the portal and address::city.2 for the second. (See “Note The -max parameter now returns 0 if the request returns no records.” on page 5-9 for information.)

Examples of –edit requests

To edit a record using a hypertext link:

```
<a href="FMPro?–db=employees.fp5&–format=–fmp_xml&
–recid=13&Country=USA&–edit">
```

To edit a record using a form action:

```
<form action="FMPro" method="post">
<input type="hidden" name="–db" value="employees.fp5">
<input type="hidden" name="–format" value="–fmp_xml">
<input type="hidden" name="–recid" value="13">
<input type="text" size=12 name="Country" value="Type a country name here">
<input type="submit" name="–edit" value="Edit This Record">
</form>
```

Generating a –delete request

Name/Value Type: Delete Record Request

What it does: Deletes the record as specified by –recid parameter. In order to delete a record, the web user must have record deleting privileges for the database.

Required parameters: –db, –format, –recid

Optional parameters: –styletype, –stylehref

Examples of –delete requests

To delete a record using a hypertext link:

```
<a href="FMPro?–db=employees.fp5&–format= –fmp_xml&–recid=4&
–delete">Delete record with ID 4</a>
```

To delete a record using a form action:

```
<form action="FMPro" method="post">
<input type="hidden" name="–db" value="employees.fp5">
<input type="hidden" name="–format" value="–fmp_xml">
<input type="hidden" name="–recid" value="4">
<input type="submit" name="–delete" value="Delete This Record">
</form>
```

Generating a –dbnames request

Name/Value Type: Database Names Request

What it does: Retrieves the names of all databases that are open and shared via the Web Companion.

Required parameters: –format

Optional parameters: –styletype, –stylehref

Examples of –dbnames requests

To retrieve the database names using a hypertext link:

```
<a href="FMPro?–dbnames &–format= –fmp_xml&–styletype=
text/css&–stylehref=mystylesheet.css”>
```

To retrieve the database names using a form:

```
<form action="FMPro" method="post">
<input type="hidden" name="–format" value="–fmp_xml">
<input type="hidden" name="–styletype" value="text/css">
<input type="hidden" name="–stylehref" value="mystylesheet.css">
<input type="submit" name="–dbnames" value="Show Database List">
</form>
```

Generating a –layoutnames request

Name/Value Type: Layout Names Request

What it does: Retrieves the names of all available layouts for a specified database that is open and shared via the Web Companion.

Required parameters: –db, –format

Optional parameters: –styletype, –stylehref

Examples of –layoutnames requests

To retrieve the names of available layouts using a link:

```
<a href="FMPro?–db=employees.fp5&–layoutnames&–format=
–fmp_xml&–styletype=text/css&–stylehref=mystylesheet.css”>
```

To retrieve the names of available layouts using a form:

```
<form action="FMPro" method="post">
<input type="hidden" name="–db" value="employees.fp5">
<input type="hidden" name="–format" value="–fmp_xml">
<input type="hidden" name="–styletype" value="text/css">
<input type="hidden" name="–stylehref" value="mystylesheet.css">
<input type="submit" name="–layoutnames" value="Show List of
Layouts">
</form>
```

Generating a –scriptnames request

Name/Value Type: Script Names Request

What it does: Retrieves the names of all available scripts for a specified database that is open and shared via the Web Companion.

Required parameters: –db, –format

Optional parameters: –styletype, –stylehref

Examples of –scriptnames requests

To retrieve the names of all scripts using a link:

```
<a href="FMPro?–db=employees.fp5&–scriptnames&–format=
–fmp_xml&–styletype=text/css&–stylehref=mystylesheet.css”>
```

To retrieve the names of all scripts using a form:

```
<form action="FMPro" method="post">
<input type="hidden" name="–db" value="employees.fp5">
<input type="hidden" name="" value="">
<input type="hidden" name="–format" value="–fmp_xml">
<input type="hidden" name="–styletype" value="text/css">
<input type="hidden" name="–stylehref" value="mystylesheet.css">
<input type="submit" name="–scriptnames" value="Show List of Scripts">
</form>
```

Generating a –dbopen request

Name/Value Type: Open Database Request

What it does: Opens a specified database that's located in the Web folder with Remote Administration enabled in the Web Companion (web users must enter "Admin" as the user name)

Required parameters: –db, –format

Optional parameter: –password

Examples of –dbopen requests

To open a remotely administered database using a link:

```
<a href="FMPro?–db=employees.fp5&–dbopen&–format=
–fmp_xml&–styletype=text/css&–stylehref=mystylesheet.css”>
```

To open a remotely administered database using a form:

```
<form action="FMPro" method="post">
<input type="hidden" name="–db" value="employees.fp5">
<input type="hidden" name="–format" value="–fmp_xml">
<input type="hidden" name="–styletype" value="text/css">
<input type="hidden" name="–stylehref" value="mystylesheet.css">
<input type="submit" name="–dbopen" value="Open Employees
Database">
</form>
```

For more information, see "Opening password-protected databases remotely" on page 3-18.

Generating a –dbclose request

Name/Value Type: Close Database Request

What it does: Closes an open database that's located in the Web folder with Remote Administration enabled in the Web Companion (web users must enter "Admin" as the user name)

Required parameters: –db, –format

Examples of –dbclose requests

To close a remotely administered database using a link:

```
<a href="FMPro?–db=employees.fp5&–dbclose=&–format=
–fmp_xml&–styletype=text/css&–stylehref=mystylesheet.css”>
```

To close a remotely administered database using a form:

```
<form action="FMPro" method="post">
<input type="hidden" name="–db" value="employees.fp5">
<input type="hidden" name="–format" value="–fmp_xml">
<input type="hidden" name="–styletype" value="text/css">
<input type="hidden" name="–stylehref" value="mystylesheet.css">
<input type="submit" name="–dbclose" value="Close Employees
Database">
</form>
```

For more information, see "Opening password-protected databases remotely" on page 3-18.

Specifying parameters for the request

The following are the parameters for requesting FileMaker Pro data in XML format. Some parameters are required to be present in the CGI command along with certain requests; others are optional.

–db (Database)

Name/Value Type: Parameter

What it does: Specifies the database that all processing for the request will refer to.

Value is: Name of the database, including the extension if any. The FileMaker Pro Web Companion uses only the name of the database; do not include any path information. The database must be open in FileMaker Pro.

Required with: All requests except the –dbnames request

Example: FMPro?–db=employees.fp5&–format= –fmp_xml&–find

–lay (Layout)

Name/Value Type: Parameter

What it does: Specifies the layout that is used in the database. The –lay parameter is used in –find, –findall, and –findany requests to specify the fields that are returned and in –view requests to specify the layout information that’s returned.

Value is: Name of the layout to use. If no layout is given, then the layout is considered to contain all fields in the database (but not related fields).

Required with: –view requests, and –edit or –new requests for data in related fields or portals

Optional with: –find, –findall, or –findany requests

Example: FMPro?–db=employees.fp5&–format= –fmp_xml&–lay= LayoutOne&–view

–format (Format)

Name/Value Type: Parameter

What it does: Specifies the XML grammar used for returning the results of a request.

Value is: –dso_xml, –dso_xml_dtd, –fmp_xml, or –fmp_xml_dtd.

Required with: All requests

Examples:

To generate this XML grammar	Specify this format
FMPDSORESULT	FMPro?–db=employees.fp5&–format= "–dso_xml"&–find
FMPDSORESULT + document type definition	FMPro?–db=employees.fp5&–format= "–dso_xml_dtd"&–find
FMPXMLRESULT	FMPro?–db=employees.fp5&–format= "–fmp_xml"&–find
FMPXMLRESULT + document type definition	FMPro?–db=employees.fp5&–format= "–fmp_xml_dtd"&–find
FMPXMLLAYOUT	FMPro?–db=employees.fp5&–format= "–fmp_xml"&–view
FMPXMLLAYOUT + document type definition	FMPro?–db=employees.fp5&–format= "–fmp_xml_dtd"&–view

–recid (Record ID)

Name/Value Type: Parameter

What it does: Defines which record should be operated on. Used mainly by the –edit, and –delete requests.

Value is: A record ID, which is a unique specifier to a record in a FileMaker Pro database.

Required with: –edit and –delete requests

Optional with: –find requests

Example: FMPro?–db=employees.fp5&–format= –fmp_xml&
–recid=22&–delete

–modid (Modification ID)

Name/Value Type: Parameter

What it does: Refers to the latest version (incremental counter number) of the record. This allows you to take necessary measures to ensure an –edit request is applied to the most current version of the record, by including a warning and an option to retrieve the most current record before the –edit request is allowed.

Value is: A modification ID, which is a unique identifier for the current version of a record in a FileMaker Pro database.

Optional with: –edit requests

Requires: The –recid parameter

Example: FMPro?–db=employees.fp5&–format= –fmp_xml&
–recid=22&–modid=6&last_name=Jones&–edit

–lop (Logical operator)

Name/Value Type: Parameter

What it does: Specifies how the find criteria are combined as either an AND or OR –find request.

Value is: Either AND or OR. If the –lop parameter name is not used, then the find request is assumed to be an AND request.

Optional with: –find requests

Example: FMPro?–db=employees.fp5&–format= –fmp_xml&
Last+Name=Smith&Birthdate=2/5/1972&–lop=and&–find

–op (Comparison operator)

Name/Value Type: Parameter

What it does: Specifies the comparison operator to apply to the field name/value pair that follows it in a –find request.

Value is: The operator to use. There are short and long versions of each operator. The default operator is "begins with". Valid operators are as follows:

Short	Long	FileMaker Pro equivalent operator
eq	equals	=word
cn	contains	"word"
bw	begins with	word*
ew	ends with	*word
gt	greater than	> word
gte	greater than or equals	>= word
lt	less than	< word
lte	less than or equals	<= word
neq	not equals	omit, word

You can use any FileMaker Pro –find operator by specifying the begins with (bw) parameter. For example, to specify the "Find Content Match" (= =) operator, you would specify the begins with parameter (bw) and then you would place the characters "==" before the actual search criteria. The required lines would look like this:

```
<input type="hidden" name="–op" value="bw">
<input type="text" name="FirstName" value="= =Sam">
```

Optional with: –find requests

Requires: A field name and a value

Example: FMPro?–db=employees.fp5&–format= –fmp_xml&
–op=eq&FirstName=Sam&–find

–max (Maximum records)

Name/Value Type: Parameter

What it does: Specifies the maximum number of records that should be returned.

Value is: A number from 1 through 2147483647, or the word "All". The default value is 25.

Optional with: –find requests

Example: FMPro?–db=employees.fp5&–format= –fmp_xml&
–max=10&–find

–skip (Skip records)

Name/Value Type: Parameter

What it does: Tells FileMaker Pro how many records to skip in the found set.

Value is: A number from 0 through 2147483647, or the word "All". If the value is greater then the number of records in the found set or the value is "All" then the last record is displayed. The default value is 0.

Optional with: –find requests

Example: FMPro?–db=employees.fp5&–format= –fmp_xml&
–skip=10&–max=5&–find

In this example, the first 10 records in the found set are skipped and records 11 through 15 are returned.

–sortfield (Sort field)

Name/Value Type: Parameter

What it does: Specifies the field that will be used for sorting. The –sortfield parameter can been used multiple times to perform multiple field sorts. The position in which –sortfield appears in the CGI command will determine the sort order of the fields.

Value is: Name of a FileMaker Pro field.

Optional with: –find or –findall requests

Example: FMPro?–db=employees.fp5&–format= –fmp_xml&
–sortfield=First+Name&–find

–sortorder (Sort order)

Name/Value Type: Parameter

What it does: Indicates the direction of a sort. If used, –sortorder must directly follow the –sortfield parameter it applies to.

Value is: The sort order. Valid sort orders are as follows, where Custom is the value list name:

Keyword (short)	Keyword (long)	FileMaker Pro Equivalent Operator
Ascend	Ascending	Sort a to z, –10 to 10
Descend	Descending	Sort z to a, 10 to –10
Custom		Sort using the value list associated with the field on the layout

Optional with: –find or –findall requests

Requires: The –sortfield parameter

Example: FMPro?–db=employees.fp5&–format= –fmp_xml&
–sortfield=First+Name&–sortorder=descend&–find

–script (Script)

Name/Value Type: Parameter

What it does: Specifies the FileMaker Pro script that will be performed after finding and sorting records (if specified) during processing of the –find request.

Value is: Name of the script to perform.

Optional with: –find or –findall requests

Example: FMPro?–db=employees.fp5&–format= –fmp_xml&
–script=Omit+Script&–find

–script.prefind (Script before Find)

Name/Value Type: Parameter

What it does: Specifies the FileMaker Pro script that will be performed before finding and sorting of records (if specified) during processing of the –find request.

Value is: Name of the script to perform.

Optional with: –find or –findall requests

Example: FMPro?–db=employees.fp5&–format= –fmp_xml&
–script.prefind=My+Script&–find

–script.presort (Script before Sort)

Name/Value Type: Parameter

What it does: Specifies the FileMaker Pro script that will be performed after finding records and before sorting records (if specified) during processing of the –find request.

Optional with: –find or –findall requests

Example: FMPro?–db=employees.fp5&–format= –fmp_xml&
–script.presort=OmitOne&–find

–styletype (Style type)

Name/Value Type: Parameter

What it does: Tells the FileMaker Pro Web Companion to generate an XML-stylesheet processing instruction within the grammar—setting the value of the type attribute (type=text/css or type=text/xsl)—so you can use cascading style sheets (CSS) or Extensible Stylesheet Language (XSL) documents with your XML document. This parameter is used in conjunction with the –stylehref parameter.

Optional with: All requests

Requires: The –stylehref parameter

Example: FMPro?–db=employees.fp5&–format= –fmp_xml&
–styletype=text/css&–stylehref=mystylesheet.css&–find

–stylehref (Style href)

Name/Value Type: Parameter

What it does: Tells the FileMaker Pro Web Companion to generate an XML-stylesheet processing instruction within the grammar—setting the value of the href attribute (href=document.css or href=document.xsl)—so you can use cascading style sheets (CSS) or Extensible Stylesheet Language (XSL) documents with your XML document. This parameter is used in conjunction with the –styletype parameter.

Optional with: All requests

Requires: The –styletype parameter

Example: FMPro?–db=employees.fp5&–format= –fmp_xml&
–styletype=text/xsl&–stylehref=mystylesheet.xsl&–find

–password (Database password)

Name/Value Type: Parameter

What it does: Specifies the database password (set in the Access Privileges > Define Passwords dialog box) to open a database with.

Optional with: The –dbopen request

Example: FMPro?–db=employees.fp5&–dbopen&–password=dbpassword&–format= –fmp_xml& –styletype=text/css& –stylehref=mystylesheet.css

field name (Name of specific field)

Name/Value Type: Field name

What it does: Field names are used to control –find criteria or to modify the contents of a record. When a value for a specific field needs to be sent to FileMaker Pro, the name portion of the name/value pair is the name of the field in the FileMaker Pro database. Field names used in this manner should not start with the hyphen (–) character.

Name is: Name of the field in the database.

Value is: For –new and –edit requests, the value contains the data for a record. Multiple occurrences of a field allow the data to be put into separate repetitions of a repeating field. For –find requests, the value is a find request on the specified field. For all other requests, these name/value pairs are not needed.

Required with: –new and –edit requests

Optional with: –find requests

Example: FMPro?–db=employees.fp5&–format= –fmp_xml& –op=eq&FirstName=Sam&–max=1&–find

Appendix B
FileMaker Pro values for error codes

The FileMaker Pro Web Companion generates an error code for databases published in XML, JDBC, or CDML format every time data is requested. The following table describes the value of each error code, which is based on the type of query to the database. Use these values to do error handling on your web pages.

For more information, see the Status (CurrentError) function described in chapter 11 of the *FileMaker Pro User's Guide* or type `error messages` in the Index tab of FileMaker Pro Help.

Note In FileMaker Pro for Mac OS, if an error occurs while performing an AppleScript from ScriptMaker, the AppleScript error code will be returned.

Error code value	Description
-1	Unknown error
0	No error (success)
1	User canceled action
2	Memory error
3	Command is unavailable (for example, wrong operating system, wrong mode, etc.)
4	Command is unknown
5	Command is invalid (for example, a Set Field script step does not have a calculation specified)
100	File is missing
101	Record is missing
102	Field is missing
103	Relationship is missing
104	Script is missing
105	Layout is missing
200	Record access is denied
201	Field cannot be modified
202	Field access is denied
203	No records in file to print or password doesn't allow print access
204	No access to field(s) in sort order
205	Cannot create new records; import will overwrite existing data
206	User does not access to import over all records
207	User does not have access to modify field definitions
300	The file is locked or in use
301	Record is in use by another user
302	Script definitions are in use by another user
303	Paper size is in use by another user
304	Password definitions are in use by another user
305	Relationship or value list definitions are in use by another user
306	Record modification ID does not match
400	Find criteria is empty
401	No records match the request
402	Selected field is not a match field for a lookup
403	Exceeding maximum record limit for trial version of FileMaker Pro
404	Sort order is invalid

Error code value	Description
405	Number of records specified exceeds number of records that can be omitted
406	Replace/Reserialize criteria is invalid
407	One or both match fields are missing (invalid relationship)
408	Specified field has inappropriate data type for this operation
409	Import order is invalid
410	Export order is invalid
411	Cannot perform delete because related records cannot be deleted
412	Wrong version of FileMaker Pro used to recover file
500	Date value does not meet validation entry options
501	Time value does not meet validation entry options
502	Number value does not meet validation entry options
503	Value in field is not within the range specified in validation entry options
504	Value in field is not unique as required in validation entry options
505	Value in field is not an existing value in the database as required in validation entry options
506	Value in field is not listed on the value list specified in validation entry option
507	Value in field failed calculation test of validation entry option
508	Invalid value entered in Find mode
509	Field requires a valid value
510	Related value is empty or unavailable
511	Field failed maximum contests validation test
600	Print error has occurred

Error code value	Description
601	Combined header and footer exceed one page
602	Body doesn't fit on a page for current column setup
603	Print connection lost
700	File is of the wrong file type for import
701	Data Access Manager can't find database extension file
702	The Data Access Manager was unable to open the session
703	The Data Access Manager was unable to open the session; try later
704	Data Access Manager failed when sending a query
705	Data Access Manager failed when executing a query
706	EPSF file has no preview image
707	Graphic translator cannot be found
708	Can't import the file or need color computer to import file
709	QuickTime movie import failed
710	Unable to update QuickTime file reference because the database is read-only
711	Import translator can not be found
712	XTND version is incompatible
713	Couldn't initialize the XTND system
714	Password privileges do not allow the operation
715	Missing Excel named range or spreadsheet
716	ODBC import should not be modifying the data source
800	Unable to create file on disk
801	Unable to create temporary file on System disk
802	Unable to open file
803	File is single user or host cannot be found
804	File cannot be opened as read-only in its current state
805	File is damaged; use Recover command

Error code value	Description
806	File cannot be opened with this version of FileMaker Pro
807	File is not a FileMaker Pro file or is severely damaged
808	Cannot open file because access privileges are damaged
809	Disk/volume is full
810	Disk/volume is locked
811	Temporary file cannot be opened as FileMaker Pro file
812	Cannot open the file because it exceeds host capacity
813	Record Synchronization error on network
814	File(s) cannot be opened because maximum number is open
815	Couldn't open lookup file
816	Unable to convert file
817	File’s binding key does not match this runtime application
818	FileMaker cannot network
900	General spelling engine error
901	Main spelling dictionary not installed
902	Could not launch the Help system
903	Command cannot be used in a shared file
904	Command can only be used in a file hosted under FileMaker Server
950	Adding repeating related fields is not supported
951	An unexpected error occurred
952	Email error message
953	Email error message
971	The user name is invalid
972	The password is invalid
973	The database is invalid
974	Permission Denied
975	The field has restricted access
976	Security is disabled
977	Invalid client IP address
978	The number of allowed guests has been exceeded (for the 10 guest limit over a 12 hour period)

This page intentionally left blank.

Appendix C
Enabling the FileMaker Pro Web Companion in Mac OS X

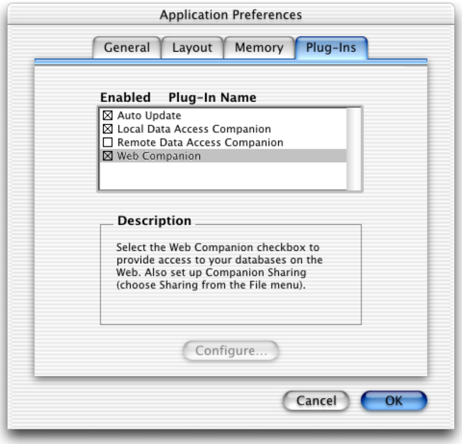
FileMaker Pro Unlimited uses the FileMaker Pro Web Companion plug-in to serve databases over the Web.

To enable the Web Companion in Mac OS X:

1. Choose the FileMaker Pro application menu > Preferences > Application.

2. In the Application Preferences dialog, click the Plug-Ins tab.

3. Select the Web Companion checkbox.



Enabled Plug-In Name

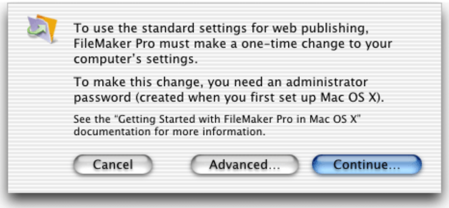
☒ Auto Update
☒ Local Data Access Companion
☐ Remote Data Access Companion
☒ Web Companion

Description

Select the Web Companion checkbox to provide access to your databases on the Web. Also set up Companion Sharing (choose Sharing from the File menu).

After you enable the Web Companion, you must specify which *port*, or virtual connection, the Web Companion will use to publish data.

The first time you enable the Web Companion (or if you have previously enabled the Web Companion and are reinstalling FileMaker Pro), FileMaker Pro requests permission to make a one-time change to your computer's setting to facilitate web publishing on ports below 1024.



To use the standard settings for web publishing, FileMaker Pro must make a one-time change to your computer's settings.

To make this change, you need an administrator password (created when you first set up Mac OS X).

See the "Getting Started with FileMaker Pro in Mac OS X" documentation for more information.

The standard port for web publishing is port number 80 (ports are numbered between 1 and 65535), and most web servers and browsers use this port as the default. Port 80 is also the default port for the FileMaker Pro Web Companion. For security reasons, Mac OS X restricts access to ports below 1024. To configure the FileMaker Pro Web Companion to use ports below 1024 while maintaining the Mac OS X access restrictions on these ports, it is necessary to make a one-time change to the file permissions of the Web Companion Enabler to give it the authority to open privileged ports (ports 1-1023). To make this change, you will need an administrator password, such as the password created when you first set up Mac OS X.

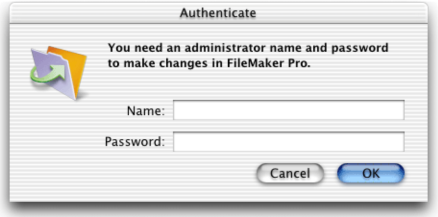
Note If you use a port other than port number 80 for FileMaker Pro web publishing, your users will need to append a colon (":") and the number of the port to their URLs to access your web hosted databases. For more information, see "Accessing databases that are published to the Web" on page C-3, and the FileMaker Pro Help topic, "Specifying a port number for web publishing."

4. If have an administrator password and want to configure the Web Companion to use standard settings for web publishing (recommended), click Continue.

If you do not have an administrator password, or do not want to enable the Web Companion at this time, click Cancel. The Web Companion will not be enabled, and your system settings will remain unchanged.

If you do not have an administrator password, or want to configure the Web Companion to use port 1024 or higher, click Advanced. Your system settings will remain unchanged. See the section "Configuring the Web Companion for use with ports 1024 and higher" for further instructions.

5. Enter an administrator name and password in the Authenticate dialog, and click OK.



You need an administrator name and password to make changes in FileMaker Pro.

6. The administrator name and password you enter can be the same as the name and password used when Mac OS X was installed, or if you have administrator privileges but do not know an administrator password, you can create a new user and password with administrator privileges. For more information on creating an account with administrator privileges, see the Mac OS X Help topics, "Working as an administrator," and "Changing your password."

You are finished. The Web Companion is configured to use port 80.

Configuring the Web Companion for use with ports 1024 and higher

You do not need an administrator password to configure the FileMaker Pro Web Companion to use ports 1024 and higher. Unlike ports below 1024, the FileMaker Pro Web Companion can use ports 1024 and above without altering your system's settings.

Note If you have previously enabled the Web Companion to use ports below 1024 as described above, your system is already configured to allow the Web Companion to use any port. Make port changes directly in the Web Companion Configuration dialog.

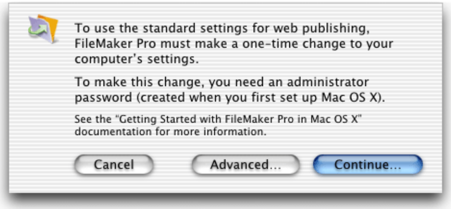
To configure the FileMaker Pro Web Companion to use only ports 1024 and above:

1. Choose the FileMaker Pro application menu > Preferences > Application.

2. In the Application Preferences dialog, click the Plug-Ins tab.

3. Select the Web Companion checkbox.

You see the following dialog.

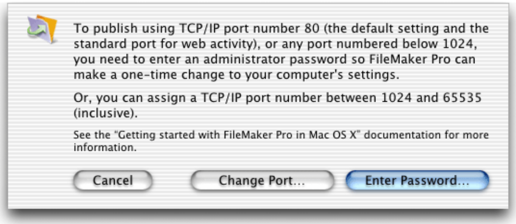


To use the standard settings for web publishing, FileMaker Pro must make a one-time change to your computer's settings.

To make this change, you need an administrator password (created when you first set up Mac OS X).

See the "Getting Started with FileMaker Pro in Mac OS X" documentation for more information.

4. Click Advanced.



To publish using TCP/IP port number 80 (the default setting and the standard port for web activity), or any port numbered below 1024, you need to enter an administrator password so FileMaker Pro can make a one-time change to your computer's settings.

Or, you can assign a TCP/IP port number between 1024 and 65535 (inclusive).

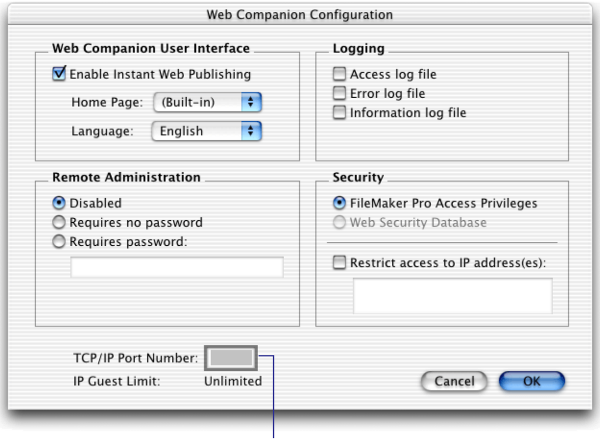
See the "Getting started with FileMaker Pro in Mac OS X" documentation for more information.

5. Click Change Port.

You see the Web Companion Configuration dialog.

Note If you click Enter Password in the above dialog, you are brought to the Authenticate dialog and asked to enter an administrator password to enable the use of ports below number 1024, as described in the previous section.

6. Enter a port number between 1024 and 65535 in the TCP/IP Port Number box.



Web Companion Configuration

Web Companion User Interface

☑ Enable Instant Web Publishing

Home Page: (Built-in)

Language: English

Logging

□ Access log file

□ Error log file

□ Information log file

Remote Administration

● Disabled

○ Requires no password

○ Requires password:

Security

● FileMaker Pro Access Privileges

○ Web Security Database

□ Restrict access to IP address(es):

TCP/IP Port Number:

IP Guest Limit: Unlimited

Enter the TCP/IP port number here

Note The IP Guest Limit should indicate Unlimited. If the IP Guest Limit does not indicate unlimited guests, you will need to reinstall FileMaker Pro 5.5 Unlimited using the installation code included with your copy of FileMaker Pro Unlimited.

7. Click OK to save your changes.

You are finished. The Web Companion is now configured to use the port you have specified.

Accessing databases that are published to the Web

When you publish a database to the Web, your users access that database by entering the host machine’s URL in their web browser. If the FileMaker Pro Web Companion is configured to use port number 80, the default port, the URL your users will enter will look like this:

```
http://12.34.56.78/
```

The number “12.34.56.78” is an example; in its place, your users would enter the actual URL of your host machine.

If the FileMaker Pro Web Companion is configured to use a port number other than the default, the URL your users will enter will look like this:

```
http://12.34.56.78:1024
```

Again, the number “12.34.56.78:1024” is an example; your users would enter the actual URL of the host machine, followed by a colon (“:”) and the port number specified in the FileMaker Pro Web Companion Configuration dialog.

This page intentionally left blank.

Index