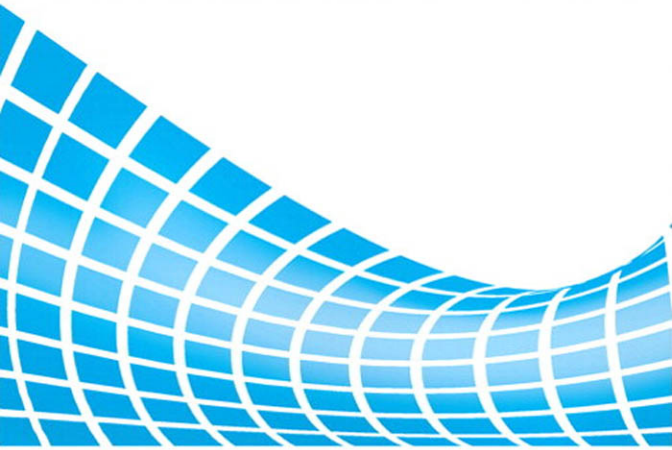


FileMaker® Pro10



IN DEPTH

FILEMAKER® PRO 10 IN DEPTH

Copyright © 2009 by Pearson Education, Inc.

All rights reserved. No part of this book shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from the publisher. No patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained herein.

ISBN-13: 978-0-7897-3946-9

ISBN-10: 0-7897-3946-1

Library of Congress Cataloging-in-Publication Data:

Feiler, Jesse.

FileMaker Pro 10 in depth / Jesse Feiler. -- 1st ed.

p. cm.

Includes index.

ISBN 978-0-7897-3946-9

1. FileMaker (Computer file) 2. Database management. I. Title.

QA76.9.D3F443 2009

005.75'65--dc22

2008054413

Printed in the United States of America

First Printing: January 2009

Trademarks

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Que Publishing cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

Warning and Disclaimer

Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied. The information provided is on an “as is” basis. The author and the publisher shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book.

Bulk Sales

Que Publishing offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales. For more information, please contact

U.S. Corporate and Government Sales

1-800-382-3419

corpsales@pearsontechgroup.com

For sales outside of the U.S., please contact

International Sales

international@pearson.com

Associate Publisher

Greg Wiegand

Acquisitions Editor

Loretta Yates

Development Editor

Kevin Howard

Managing Editor

Patrick Kanouse

Project Editor

Mandie Frank

Copy Editor

Chuck Hutchinson

Indexer

Tim Wright

Proofreader

Kathy Ruiz

Technical Editor

Andrew Knasinski

Publishing Coordinator

Cindy Teeters

Designer

Anne Jones

Compositor

TnT Design, Inc.

WELCOME TO FILEMAKER PRO 10

BEST OF THREE WORLDS

Welcome to the world of FileMaker Pro 10. By simply browsing through this book, you're sure to have seen the word *database*. We cover what databases are in the rest of this book, but one of the first things you need to understand about FileMaker Pro 10 is that it is far more than just a database application.

FileMaker Pro 10 is nearly unique in the world of software. It is a powerful database system that can manage and store a wide range of information—it's an application for end users (like Microsoft Excel or Intuit's Quicken), and it's also a robust rapid application software development platform.

When you hear people speak about FileMaker, keep in mind they might be viewing it from any one of these different perspectives. An IT professional likely sees FileMaker as a database engine that fits into a larger security and network infrastructure. An end user is probably thinking about a specific solution built in FileMaker Pro and how it helps make her work more efficient. A software developer might see FileMaker as one of many tools he employs in building a wide range of applications.

This book was written with an eye toward the FileMaker developer community. If you're mostly interested in learning how to use the essential features of the FileMaker application, though, this book might not be for you. Although we've included some introductory chapters to be as comprehensive as possible, we've chosen to focus on an audience that we assume is largely familiar with the essential operations of FileMaker already and is interested mostly in topics for the beginning to advanced developer.

HOW THIS BOOK IS ORGANIZED

FileMaker Pro 10 In Depth is divided into five parts, organized into something like a tree. Part I, "Getting Started with FileMaker 10," and Part II, "Developing Solutions with FileMaker," constitute the "trunk" of the tree; they cover fundamental material that we recommend everyone read.

Subsequent parts branch out from this base. Part III, “Developer Techniques,” focuses on using FileMaker’s features to develop complete, robust database applications. Part IV, “Data Integration and Publishing,” covers getting data into and out of FileMaker. And Part V, “Deploying a FileMaker Solution,” covers options for making a FileMaker solution accessible to others.

The following sections describe the five parts of *FileMaker Pro 10 In Depth* and the topics they cover.

PART I: GETTING STARTED WITH FILEMAKER 10

The chapters in Part I introduce you to FileMaker and its uses and features, and get you started with the basics of defining databases.

- Chapter 1, “FileMaker Overview,” situates FileMaker Pro within the wider world of database and productivity software. It provides an overview of the new FileMaker 10 product line and mentions the most important new features in FileMaker 10. This chapter is appropriate both for those who are new to FileMaker Pro and for those who have used previous versions and want a quick tour of the major innovations.
- Chapter 2, “Using FileMaker Pro,” is intended as an introduction to the software from the perspective of a database user rather than a database developer. We introduce the major components and functions of the FileMaker interface, such as the status area, layouts, FileMaker’s modes, and the basics of record creation, editing, and deletion.
- Chapter 3, “Defining and Working with Fields and Tables,” provides a thorough overview of all of FileMaker’s field types and field options, including lookups, validation, storage types, and indexing. This chapter is intended to help lay the groundwork for talking about database development and to serve as a thorough reference on FileMaker field types and options.
- Chapter 4, “Working with Layouts,” covers all of FileMaker’s layout-building options in detail. We cover all aspects of layout building and offer guidelines for quicker and more efficient layout work.

PART II: DEVELOPING SOLUTIONS WITH FILEMAKER

Part II is intended to introduce you to the fundamental techniques of database application development using FileMaker Pro and FileMaker Pro Advanced. Chapters 5 through 7 cover the theory and practice of designing and building database systems with multiple data tables. Chapters 8 through 10 introduce you to foundational concepts in application and reporting logic.

- Chapter 5, “Relational Database Design,” introduces you to relational database design concepts. We proceed by working on paper, without specific reference to FileMaker, and introduce you to the fundamental vocabulary and techniques of relational database design (keys and relationships).
- Chapter 6, “Working with Multiple Tables,” begins the task of translating the generic database design concepts of Chapter 5 into specific FileMaker techniques. We show how to translate a paper diagram into an actual FileMaker table structure. We show how to model different relationship types in FileMaker using multiple data tables and how to create fields that function effectively as relational keys.

- Chapter 7, “Working with Relationships,” builds on the concepts of Chapter 6. Rather than focusing on FileMaker’s relationships from the standpoint of database design, we focus on their practical implementation in FileMaker programming. We look in detail at the new capabilities of FileMaker 10 and discuss nonequality join conditions, file references, and some strategies for organizing a multitabled system.
- Chapter 8, “Getting Started with Calculations,” introduces FileMaker’s calculation engine. The chapter delves into the major types of FileMaker calculations. We cover a number of the most important functions and discuss general strategies and techniques for writing calculations.
- Chapter 9, “Getting Started with Scripting,” introduces FileMaker’s scripting engine. Like the preceding chapter, this one covers the fundamentals of an important skill for FileMaker developers. We cover some common scripting techniques and show how to use event-driven scripts to add interactivity to a user interface.
- Chapter 10, “Getting Started with Reporting,” illustrates the fundamental techniques of FileMaker Pro reporting, such as list views and subsummary reports, as well as some more advanced subsummary techniques, and some design techniques for improving the look and usability of your reporting layouts.

PART III: DEVELOPER TECHNIQUES

The chapters in Part III delve deeper into individual topics in advanced FileMaker application development. We build on earlier chapters by exploring more complex uses of portals, calculations, and scripts. We also offer chapters that help you ready your FileMaker solutions for multi-user deployment, and we examine the still-important issue of conversion from previous versions.

- Chapter 11, “Developing for Multiuser Deployment,” explores the issues and challenges of designing FileMaker systems that will be used by several people at once. We discuss how FileMaker handles concurrent access to data and discuss the concept of user sessions.
- Chapter 12, “Implementing Security,” is a thorough overview of the FileMaker 10 security model. We cover the role-based accounts feature, extended privileges, and many of the complexities of server-based external authentication against Windows or Mac OS X user directories, for example.
- Chapter 13, “Using the Web Viewer,” explores one of the interesting recent features of FileMaker Pro. You can incorporate live web pages into your FileMaker layouts, and you can use data from the FileMaker database to construct the URLs that are displayed.
- Chapter 14, “Advanced Interface Techniques,” provides detailed explanations of a number of more complex, applied techniques for working with layouts and data presentation in a FileMaker application.
- Chapter 15, “Advanced Calculation Techniques,” looks closely at some of the more advanced or specialized types of FileMaker calculations, as well as the functions for text formatting and for list manipulation.
- Chapter 16, “Advanced Scripting Techniques,” like the preceding chapter, is full of information specific to features of FileMaker 10 scripting. Here, we cover programming with script parameters, the significant feature of script variables, programming in a multiwindow system, and the complexities of scripted navigation among multiple tables and recordsets.



- Chapter 17, “Working with FileMaker Triggers,” examines one of the most important new features of FileMaker Pro 10. Triggers let you set up automatic behaviors that occur whenever certain events happen. They let you exercise more control over the user interface with less programming in many cases.
- Chapter 18, “Advanced FileMaker Solution Architecture,” is the last of the chapters in the Advanced series. It presents a variety of features and solutions that integrate and expand some of the techniques in the previous chapters. You will find information on window management, multiwindow interfaces, and selection portals, among other topics.
- Chapter 19, “Debugging and Troubleshooting,” is a broad look at how to find, diagnose, and cure trouble in FileMaker systems—but also how to prevent it. We look at some software engineering principles that can help make systems more robust, and can reduce the incidence and severity of errors. The chapter also includes detailed discussions of how to troubleshoot difficulties in various areas, from multiuser record lock issues to performance difficulties over large networks.
- Chapter 20, “Converting Systems from Previous Versions of FileMaker Pro,” explores the complex issues involved in moving to FileMaker 10 from versions prior to FileMaker 7. We then discuss the mechanics of conversion in detail, and discuss some of the more significant pitfalls to be aware of.

PART IV: DATA INTEGRATION AND PUBLISHING

Part IV covers technologies and capabilities that allow FileMaker to share data, either by exchanging data with other applications, or by exporting and publishing data, for example, via ODBC, JDBC, and the Web.

- Chapter 21, “Connecting to External SQL Data Sources,” explores FileMaker’s ODBC/JDBC interface as well as the exciting features that let you add SQL tables to your Relationships Graph. This means that you can now use SQL tables very much as if they were native FileMaker tables. You can use them in layouts along with FileMaker tables, you can use them in reports, and you can even expand them by adding your own variables to the FileMaker database that are merged with the external SQL data as you use it.
- Chapter 22, “Importing Data into FileMaker Pro,” looks at almost all the means by which you can import data into FileMaker. It covers how to import data from flat files, how to batch imports of images and text, and how to import images from a digital camera. (XML importing is covered in Chapter 24.) It also shows you how to import data from Bento on Mac OS X.
- Chapter 23, “Exporting Data from FileMaker,” is in some respects the inverse of Chapter 22. It covers almost all the ways by which you can extract or publish data from FileMaker.
- Chapter 24, “Instant Web Publishing,” looks at the features of the FileMaker 10 Instant Web Publishing model. Anyone interested in making FileMaker data available over the Web should begin with this chapter.
- Chapter 25, “Custom Web Publishing with XML/XSLT,” covers the first set of FileMaker 10’s Custom Web publishing technologies. It introduces you to FileMaker’s

XML capabilities and describes XML and its companion technology XSLT as they relate to FileMaker's XML import and export capabilities. You will see how to configure the FileMaker Web Publishing Engine (WPE), and how to write XSLT stylesheets that exploit the WPE's capabilities to build FileMaker-backed web applications. With XML and XSLT transformations, you can publish FileMaker data not only onto a website but also into other formats. You will see how to use the XML/XSLT Site Assistant to easily build Custom Web Publishing sites.

- Chapter 26, “Custom Web Publishing with PHP,” shows you how to use FileMaker's newest web publishing tools to build a PHP-based site. For many people, PHP sites are easier to develop and integrate into existing websites than XML/XSLT sites.

PART V: DEPLOYING A FILEMAKER SOLUTION

Part V delves into the choices you have for how to deploy a FileMaker database, including deployment via FileMaker Server and via kiosk or runtime mode using FileMaker Developer.

- Chapter 27, “Deploying and Extending FileMaker,” provides an overview of the ways you can deploy a FileMaker database to one or more users, reviews plug-ins, and explores means of distributing standalone databases. Read this chapter for a quick orientation toward your different deployment choices.
- Chapter 28, “FileMaker Server and Server Advanced,” explores in depth setting up and working with FileMaker Server and FileMaker Server Advanced. The chapter covers setting up, configuring, and tuning FileMaker Server, as well as managing server-side plug-ins and authentication. The new Server Admin Console is described in detail here.

SPECIAL FEATURES

This book includes the following special features:

- **Troubleshooting**—Many chapters in the book have a section dedicated to troubleshooting specific problems related to the chapter's topic. Cross-references to the solutions to these problems are placed in the context of relevant text in the chapter as Troubleshooting Notes to make them easy to locate.
- **FileMaker Extra**—Many chapters end with a section containing extra information that will help you make the most of FileMaker Pro. In some cases, we offer expanded, fully worked examples of tricky database design problems. In others, we offer shortcuts and maintenance techniques gleaned from our collective experience with developing production FileMaker systems (creating custom function libraries or getting the most out of team development). And in still others, we delve all the way to the bottom of tricky but vital FileMaker features such as the process of importing records.
- **Notes**—Notes provide additional commentary or explanation that doesn't fit neatly into the surrounding text. You will find detailed explanations of how something works, alternative ways of performing a task, and other tidbits to get you on your way.

- **Tips**—This feature identifies some tips and tricks we’ve learned over the years.
- **Cautions**—Here, we let you know when there are potential pitfalls to avoid.
- ★ **New in This Version icon**—This icon identifies things that are new in FileMaker 10.
- **Cross-references**—Many topics are connected to other topics in various ways. Cross-references help you link related information together, no matter where that information appears in the book. When another section is related to one you are reading, a cross-reference directs you to a specific page in the book on which you can find the related information.
- **FileMaker Scripts**—Numerous examples of scripting are provided in the book. Because you can create long lines of code, they are sometimes split in order to be printed on the page. The ➤ indicates the continuation of the previous line of code.

DOWNLOADABLE FILES

Most of the examples in this book are based on the FileMaker Starter Solutions that are installed automatically for you when you install FileMaker. Thus, you already have most of the files. In some cases, additional files or additional code has been added to the Starter Solutions as described in this book. These files can be downloaded from filemakerindepth.com. You can also download them from the publisher’s website at <http://www.informit.com/title/9780789739469>.

WHO SHOULD USE THIS BOOK

Like FileMaker itself, this book has several audiences. If you work with structured data a lot (Excel spreadsheets, for example) but are new to databases, this book will provide you with a solid foundation in the world of databases, in the basics of database theory, and in the practical skills you need to become a productive database user or developer. The book’s more introductory chapters tell you what you need to know to get started building basic databases for your own use. Later chapters introduce you to the world of multiuser database design and to some of FileMaker’s more advanced application design features.

If you’ve worked with other database systems—either server-side relational database engines based on SQL or desktop development environments such as Access—this book will help you see how FileMaker Pro fits into the universe of database software. Refer to the “How This Book Is Organized” section earlier in this Introduction to get a sense of which chapters will get you started quickly with FileMaker.

And in case you’re an old hand with FileMaker, we’ve provided a good bit of in-depth discussion of advanced techniques and have called out new FileMaker 10 features throughout the book.

DEFINING AND WORKING WITH FIELDS AND TABLES

WORKING UNDER THE HOOD

Fields and tables are the heart of any database. By storing information in properly categorized fields within well-organized tables, you impart both function and meaning to what would otherwise be an incomprehensible pile of raw data.

This chapter describes what kinds of fields exist in FileMaker Pro, how they store information, and how to ensure proper data integrity in your database solutions. We also discuss naming conventions for fields and tables—techniques that you can use to make your FileMaker Pro databases meaningful to yourself and others for the long period of time that they may be in use.

If you're new to development in FileMaker Pro, this chapter is a good place to start. Establishing a solid foundation in field definition is a vital part of becoming a practiced developer.

NEW DATABASES BEGIN WITH FIELD DEFINITIONS

To create a new database, simply launch FileMaker Pro and then choose File, New Database. The Quick Start screen appears, and you can choose the Create Database view to get started. At that point, you can choose to create an empty database or to create a database from a Starter Solution.



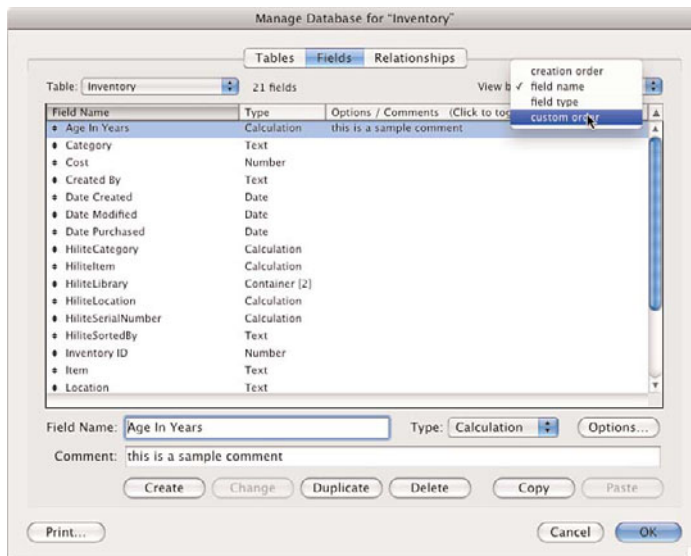
In FileMaker Pro 10, the Quick Start screen also lets you choose to create a database from an Excel workbook, a tab-delimited text file, a comma-separated values text file, a merge file, or a Bento source.

USING THE MANAGE DATABASE DIALOG

When you choose to start a new, empty database, FileMaker Pro creates a file for you, stores it in a location you specify, and automatically opens the Manage Database, dialog shown in Figure 3.1. As a developer, you'll spend a good bit of time in the three tabs in this

dialog. FileMaker Pro's Manage Database dialog allows you to create the fields, tables, and relationships you need to form your database. It also enables you to modify a wide range of attributes associated with fields, such as auto-entry functions, validation, storage, and calculation formulas. These elements compose a database's structure or *schema*. It is here that you form your database behind the scenes.

Figure 3.1
The three tabs allow you to switch among defining tables, fields, and relationships.



If you are creating a new database from a Starter Solution or an existing file such as an Excel workbook or a text file, FileMaker Pro automatically creates the necessary schema and opens the new database without going through the Manage Database dialog. While you are working with a database, you can open the Manage Database dialog at any time to modify the schema.

FileMaker Pro will have already created a default table for you, named the same as the file itself. Notice the Table pop-up menu on the Fields tab of the dialog in Figure 3.1. Any fields you create will be created in that table.

- ➔ For some basic information on tables, **see** “Understanding Tables,” **p. 30**.
- ➔ For a detailed discussion of multiple-table solutions, **see** Chapter 6, “Working with Multiple Tables,” **p. 159**.

Notice the third tab in the Manage Database dialog: Relationships. We don't cover relational databases in this chapter, but it is on that tab that you would create the relational associations among tables in your solution.

- ➔ For information on relational data modeling, **see** Chapter 5, “Relational Database Design,” **p. 143**.

WORKING WITH TABLES

As you saw in the preceding chapter, your database consists of tables, each of which is made up of rows or records with columns or fields that contain the data. A database can consist of a single table or a number of tables.

By default, when you create a new database, a single table is created that has the same name as the database. That actually might not be what is best. You might want to rename that default table so that it fits into the naming convention of all the tables in your database.

TABLE NAMING CONVENTIONS

The Manage Database dialog lets you create and name (and rename) fields and tables. It is a good idea from the start to enforce some naming conventions on both fields and tables.

FileMaker Pro's flexibility with regard to things such as legal characters in names and the length of names for tables and fields can be too much of a good thing. You can use up to 100 characters in a name, but chances are you will need far fewer for your actual names.

Here are some suggestions based on conventions used by various FileMaker developers. There is more information in the Support area of the FileMaker website; in addition, FileMaker's TechNet membership gives you access to still more information and guidelines. Pick what are the most useful conventions, but stick with them.

Stick with them, that is, within a single database or even a project. One problem with implementing design conventions is that the world is a large place, and it is likely that your naming conventions will need to interact with naming conventions of other systems and databases. Being internally consistent keeps your own house in order. That is the most that you can hope for, unless you volunteer to serve on a committee that drafts conventions for your organization, industry, or other group.

Naming tables is simultaneously simple and almost irrelevant. The reason is that as soon as you have a database with more than one table in it, you will most likely be using the Relationships graph (described in Chapter 7, "Working with Relationships"). The Relationships graph initially shows each table with the name that you assign to it. However, you will create additional instances of your tables in the Relationships graph, and you will name each of them. In practice, you will usually be working not with the base table, but with the additional instances.

For example, you might have a table called Personnel. In the Relationships graph, you might have instances of this table called PersonnelByID, PersonnelByName, PersonnelByDepartment, and so forth. Practically, you could name the base table Table 1, and, as long as the other names appear in the Relationships graph (and in your code), everything would be clear (but this is presented only as a hypothetical example, not a good practice).

When you create a database, by default you will wind up with a database, a single table, and an instance in the Relationships graph all with the same name. Many people begin by renaming that first table right away. Here are some of the suggested standards:

- Use only the characters 0–9 and a–z (both uppercase and lowercase).
- If table names contain several words, separate them with underscores or with intermediate capitalization (as in `personnelSalaryInfo` or `personnel_salary_info`).
- Be consistent in capitalization and number (that is, use table names such as `Contacts` or `Contact`, `contacts` or `contact`).
- Do not use special characters or reserved words in table names. Reserved words include FileMaker reserved words as well as words that might be reserved in SQL or other languages you can use to access the tables. *Select* is not a good table name because, although it might be useful for storing selection values for records in your database, it is an SQL reserved word.

In addition, consider whether you want to place any descriptive information in the table name. If you do so, the usual convention is to place it at the end following an underscore. This approach is particularly useful if you separate words within the table name using intermediate capitalization. For example, `inventorySuppliers_pub` and `inventory_Quantities_pri` are reasonable names for inventory tables that, respectively, contain the publicly available names and addresses of suppliers and the private quantities of inventory items on hand. You can enforce access to these tables with your security accounts and privileges, but it can be useful to indicate not only what is in the tables but also the sensitivity of the data.

CREATING NEW TABLES

To create a table, go to the Manage Database dialog (File, Manage Database). Click the Tables tab to show the view shown in Figures 3.2 and 3.3. Note that this is one of the places in which FileMaker Pro and FileMaker Pro Advanced differ.

Figure 3.2
Use the Tables tab in Manage Database to create, change, and delete tables in FileMaker Pro.

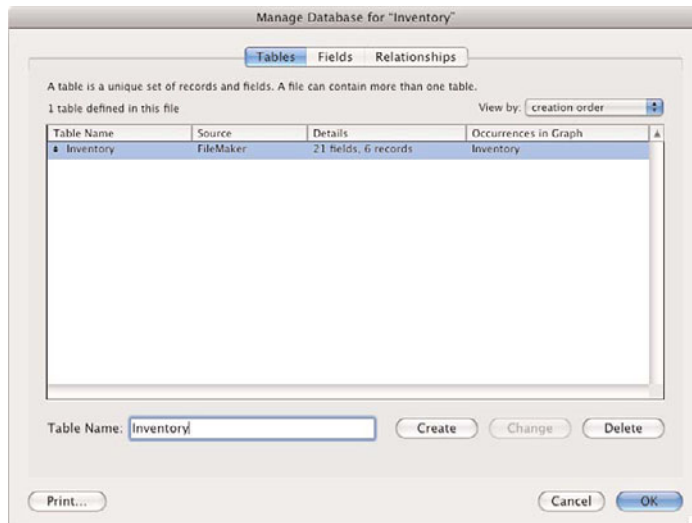
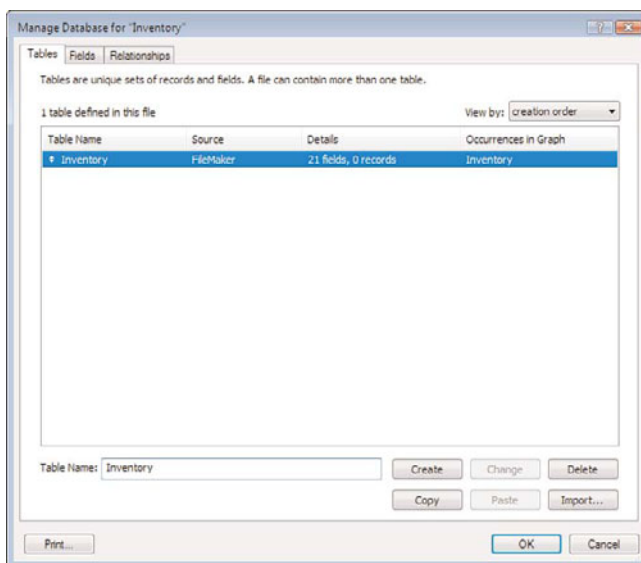


Figure 3.3

Use the Tables tab in Manage Database to create, change, and delete tables in FileMaker Pro Advanced.



To create a new table, enter a name for the table at the bottom. Click Create, and your table will be created in the list of tables. An instance of the table will also be created automatically in the Relationships graph. To rename a table, highlight its name in the list of tables in the Tables tab, type in the new name at the bottom, and click Change.

To delete a table, highlight its name and click Delete. If you want to print out the fields in one or more tables, highlight it (or them) and click Print.

NOTE

The Manage Database dialog in FileMaker Pro Advanced has three additional buttons in the lower-right corner, as shown in Figure 3.3. You can copy a table and paste it into your database (thereby duplicating it), or copy and paste it into another database. You can also import a table definition, not the data, from another FileMaker Pro database. In addition, you can import data along with the table definition, but that is done with the Import command, described in Chapter 22, "Importing Data into FileMaker Pro."

WORKING WITH FIELDS

The heart of the database is the data within it, data that is stored in fields. This section provides some basics about working with fields.

FIELD NAMING CONVENTIONS

The naming conventions for tables with regard to spaces, characters, capitalization, and so forth apply also to fields. There are some additional considerations when it comes to naming fields. Specifically, they have to do with the identification of field types and the naming of internally used fields.

Many developers use abbreviations for data types in field names. Often it's handy to know the data type of a given field when working with it without having to refer to the Manage Database dialog. Here we've used "t" for text, "n" for number, and "c" for calculation:

- ProductName_t
- Price_n
- TaxRate_n
- Tax_c

Some developers also note whether a field is indexed ("x" for indexed, "n" for unindexed):

- Location_Name_tx
- Location_Desc_tn
- Location_Size_nn

Some naming conventions also break out a division between data fields and what are commonly referred to as *developer fields*—those fields that you need only to make your FileMaker Pro solution work. If you ever went to import your database wholesale into another system, you would probably leave behind the developer fields. Here, we have two abbreviations: "k" for key (or match field) and "zz" (so that it sorts to the bottom of the list) for developer utility fields. We also use underscores to ensure that keys sort to the top of our field list, with the primary key coming first.

➔ To understand how keys are used to identify records in tables and form relationships, **see** Chapter 5, "Relational Database Design," **p. 143**.

- __kp_primary_AlbumID
- _kf_foreign_ArtistID
- AlbumName
- Date
- zz_SelectedPortalRow
- zz_UserColor_Preference
- zz_UserGenre_Preference

Many developers use a minimal set of field-naming standards. It relies on leading lowercase characters to indicate the field type. If you choose to use that minimal set, here are the conventions used:

- g—Global
- c—Calculation
- s—Summary
- zz—Internal use (This causes the field name, when shown in an alphabetical list, to be at the bottom. If you use a single "z," your internal fields will be interspersed with fields such as ZIP code.)

Descriptions of field types might or might not use this set of standards, which you can add to the end of the field name following an underscore:

- **t**—Text
- **n**—Number
- **d**—Date
- **ts**—Timestamp
- **tm**—Time
- **c**—Container

Putting these together, you could have field names such as these:

- `creationDate_d`
- `gProcessingOffice_t`
- `gcNextInvoiceNumber_n`

You can even go further by not bothering with field types where the field name already includes it. `creationDate_d` really adds no information to `creationDate`.

Whatever you do, be consistent. The point is not to create a set of naming conventions that overshadows the database but, rather, to create naming conventions that help you and future developers build and maintain the solution.

TIP

Don't imagine that all the fields on your Relationships Graph will adhere to these naming conventions. You control your own fields, but as you begin to use external data sources, you will be incorporating fields from other databases. You can have a field called `payrollDate_d` in your own table, but if you are relating it to a field called `datePaid` in the corporate database, chances are slim that the database administrator will want to rename the field to make it consistent. The `ProjectID` field in your database might be related to a field that is called `JobNumber` in another FileMaker database that you do not control. And, in a global world, the external data source names might just be in another language. Be as clear and consistent as you can, but do not assume that you can control the names of fields in other databases. (In general, the owner of the Payroll database wins out.)



If you're planning on using FileMaker Pro as a web back end, refer to "Problematic Field Names" in the "Troubleshooting" section at the end of this chapter.

- For more information on using databases on the Web, see "Designing for IWP Deployment," **p. 532**, as well as Chapter 25, "Custom Web Publishing with XML/XSLT," **p. 545**.

ADDING FIELD COMMENTS

Notice also that you can add comments to your field definitions. (A field comment was shown previously in Figure 3.1.) Commenting is a vital discipline to develop. Spending a few moments to add information to the Comment text box, below the field name, as you create a field will save time later in trying to figure out what you were thinking at the time.

Don't bother repeating information that is in the field name. If the field represents pixels or pennies, adding that information to the field name might be worthwhile (as in `Width_In_Pixels`). Use comments for in-progress remarks (such as "Added 4/1/2007 JF for task restructure" or "for reporting only").

TIP

You can extract field comments using the `FieldComment` function so that you can use them in a tooltip or other dynamic documentation in your solution.

CREATING NEW FIELDS

To create fields in FileMaker Pro, you need to enter some text in the Field Name area of the Manage Database dialog and click Create. One important aspect of databases to keep in mind is that it's important to establish a discrete field for each bit of information you want to store. If you create a field called Contact Information and cram an entire address and a set of phone numbers into it, technically it will work fine. But if you ever need to export that information, sort by area code, or run a report by city, you won't be able to cull the information you want from the field without suffering a good headache.

- To database wonks, the Contact Information example would be a violation of first normal form, or more colloquially, "one fact, one field." For information on relational data modeling and defining fields, **see** "Relationship Types," **p. 149**.

As shown previously in Figure 3.1, the Manage Database dialog lets you create, change, duplicate, and delete fields. As with the Tables tab, FileMaker Pro Advanced has additional buttons: Copy and Paste. If you select a field or fields, you can click Copy and paste the fields into the same or another table. Pasting them into the same table is the same as duplicating them.

WORKING WITH FIELD TYPES

Some of the most important aspects of understanding FileMaker Pro involve comprehending field types, realizing how they differ from one another, and knowing how to use them effectively. Simply stated, field types identify what kind of information each field of your database expects to hold. A person's name is text, the purchase amount for a transaction is a number, a birthday is a date, and so on. Generally, it should be quite clear to you what each needs to be.

Field types determine what types of operations can be performed on a given field, what information a field can accept, and the rules by which a field is sorted. The combination of a proper identifying field name and a data type definition is what gives a database its context and meaning.

TIP

Use the most specific field type you can. This allows you to use FileMaker Pro editing and formatting. Although FileMaker can convert a text field to a number where necessary, it can apply numeric formatting only to a number field. The same goes for dates and times.

TEXT

Text fields are the most free-form of the field types. Users can enter any range of information in them, including carriage returns, and there's no expectation of what form or sort of information a text field will hold. The only requirement is that it be character based; in other words, you can't place a picture in a text field. A text field can store up to 2GB of information, limited by RAM and hard drive space, of course, and indexes up to approximately 100 characters, depending on what language you're using. We cover indexing in more depth later in the chapter. For now, simply remember that each field type has different limits and approaches on indexing.

NUMBER

Number fields can store values from 10^{-400} up to 10^{400} , and negative values in the same range. FileMaker Pro indexes the first 400 significant digits (numbers, decimal points, or signs) of a number field, ignoring letters and other symbols. Number fields can accept text (although not carriage returns), but any text in a numeric field is ignored. FileMaker interprets 12ax3 as 123 if you enter it into a numeric field, for example.

Something to keep in mind with FileMaker Pro: You can express a number field as a Boolean. A Boolean value is either true or false, and often used to test the condition of something. FileMaker Pro treats a zero or null value in a number field as false in the Boolean sense; it treats any other data as true. You will often run across the use of number fields to store Boolean values.

The primary distinction between a number field and a text field lies in how they sort: A text field sorts 1, 10, 2, 20, 3, 4, 5, whereas a number field sorts 1, 2, 3, 4, 5, 10, 20.

DATE

Date fields accept only Gregorian calendar dates. FileMaker Pro honors whatever date formatting your country follows by taking the standard your operating system uses at the time you create a new file. Date formats—the order of year, month, and day—are common for a given file. Although it's possible to change the way FileMaker Pro displays dates, it fixes basic ordering at the time of file creation.

Dates in FileMaker Pro are internally stored as the number of days since 01/01/0001. January 1, 2004, for instance, is 731581. If you need to compare dates or perform any functions on them, remember that behind the scenes they're really just numbers. This feature is actually quite handy. To switch a date to a week prior, all you need to do is subtract 7. Date fields can store values from January 1, 0001, to December 31, 4000.



If your fields are sorting or displaying oddly, see “Mismatched Data Types” in the “Troubleshooting” section at the end of this chapter.

TIME

Time fields hold HH:MM:SS.ddd information. Notice that you can add a decimal to the end. An additional useful fact: If a user enters **25:00**, FileMaker Pro rightly interprets this as

1:00 a.m. 99:30 becomes 3:30 a.m. The clock simply keeps rolling over. This behavior is useful when you need to add, say, 30 hours to a time and don't want to be bothered with calculating what hour that becomes. Likewise, if you are doing data entry in a time-tracking system and don't want to create two entries for a case in which you worked from 2:00 p.m. until 2:00 a.m. on Monday (really Tuesday), entering 26:00 for the ending time in your system rightly calculates to 12 hours.

As in dates, FileMaker Pro stores time internally as the number of seconds from 12:00:00 on the current day: 1 is 12:00:01, and 43200 is 12:00 p.m. As it does with date formats, FileMaker Pro establishes your time format during the creation of the file, based on system operating system settings.

The maximum time value you can store in a FileMaker Pro time field is 2,147,483,647. That's a lot of time.

TIMESTAMP

The timestamp data type combines date and time information. It appears as a field with both date and time values, separated by a space: 1/1/2004 12:00:00. As in date and time formats, timestamps are also stored as numbers: the count of seconds from 1/1/0001 00:00:00. Be prepared to work with large numbers when using this field type. Timestamps are an important aid to interoperability with other databases (such as those powered by the SQL language), which often store date and time information in a single timestamp field. The maximum value of a timestamp is 12/31/4000 11:59:59.999999 p.m. or 126,227,764,799.999999 seconds.

TIP

To extract just the date from timestamp data, simply use the `GetAsDate()` function. Likewise, use `GetAsTime()` to extract just the time. In a layout, you can format a timestamp as a date or as a time (as well as leaving it as a timestamp). If you format a timestamp as a date, the time value is not shown in the layout.

CONTAINER

Container fields are different from the ones already mentioned: They store binary information. Information is often inserted into container fields rather than being entered manually (you can copy and paste). You can place any sort of digital document in your database, limited again by the practical limits of your computer hardware, up to 4GB.

Container fields also support displaying/playing three native types of media: pictures, QuickTime movies, and sounds. Refer to the FileMaker help system for supported formats, but most common image formats are included...as well as some you won't expect. For example, if you use QuickTime, it's possible to display and play a Macromedia Flash 5 .swf file. Last, on Windows, container fields support a wide range of OLE objects, including Microsoft Excel documents, PDF, and more. (The first page of PDF documents placed in containers will always be rendered on both operating systems but for this to happen, the document must be inserted on Mac OS X as a picture or file and on Windows must be inserted as an object.)

There's one important point to remember about using container fields: Either you can store the file or media in FileMaker itself, requiring disk space, or you can simply store a path reference to the file instead. If you choose to store just a reference to the file, FileMaker Pro displays the image or file icon as necessary, but it does not hold the actual document. A nice feature of storing references is that you can double-click documents in your container fields to launch them in your operating system.

CAUTION

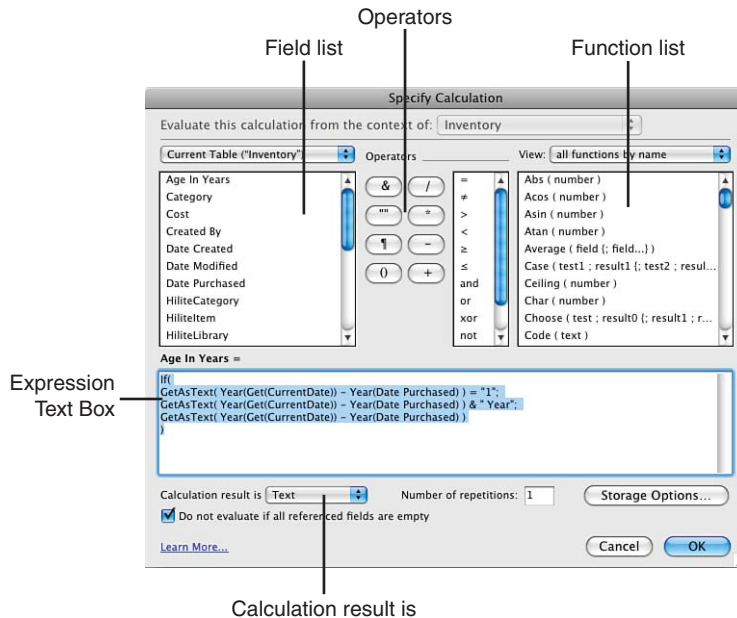
Keep in mind that if you move the source document, the FileMaker Pro reference remains but is no longer valid.

CALCULATION

Calculations evaluate formulas and return the requisite results. When you create a calculation field, the Specify Calculation dialog opens, as shown in Figure 3.4. You use the same dialog to specify calculations used for script parameters, web viewers, security privileges, and other purposes in FileMaker Pro.

3

Figure 3.4
Calculations form an essential part of FileMaker Pro development.



Features of the Specify Calculation dialog include the following:

- **Field list**—Select fields to include in your calculation from the list below the table menu. Use the drop-down menu to change from table to table. Note that double-clicking inserts a field into your calculation where your cursor currently sits.
- **Operators**—Use these buttons to insert math and special operators.

- **Function list**—Just below the View drop-down menu is a list of functions. Here, you're able to scroll through all of FileMaker Pro's various functions and then double-click to insert. It's a good idea to start here to get your syntax correct. The menu above enables you to filter your list by category to show the functions you need.
- **Expression text box**—This is the place where you assemble your actual formula or expression. This is a simple text entry area: If you want, work in a text editor and paste calculations here.
- **Calculation Result Is list**—Calculations return varying information, depending on what data/field type is required. If you want the field to be sortable by alphabet, set the return data type to Text. If you have a field returning, say, a price, set the type to Number.
- **Learn More link**—You can find more information by using this link in the lower left of the dialog.

Examples of calculations include the following:

- `3 + 4` always displays its result of 7.
- `Sale + Tax` displays the sum of two fields named Sale and Tax.
- `Personnel::EmployeeID` displays the value of a field in a related table. This type of calculation is sometimes utilized to create a field in a table that takes part in a sort or other routine where you cannot use a related field. In old FileMaker Pro databases, relationships cannot be used more than one table away. Calculations designed simply to provide an in-table copy of a related value frequently litter such databases.
- `Position (Notes; "a"; 1; 1)` returns a numeric position, starting from the first character in the field Notes, for the first "a" found.
- `IsEmpty (MyField)` returns a zero or one (Boolean) depending on whether MyField has a value in it, including zero. If a zero is entered, the field is technically not empty. Only a null value is considered empty.
- `If (MyDate > 900; "yes" ; "no")` displays a yes for dates entered in MyDate greater than 6/19/0003; otherwise, it displays no (remember that you just tested for the number of days past 1/1/0001).

You can use the Specify Calculation dialog to create a calculation just by clicking fields, operators, and functions. However, you can also type directly into the expression text box. As Figure 3.4 shows, you can spread out your calculation; spaces do not matter except within quotation marks. You can also use indentation to clarify the calculation. Comments can be inserted using two slashes (`//`), which mean that the remainder of the line is ignored. Multiline comments can be entered starting with `/*` and ending with `*/`.

➔ For more detail on calculations, **see** Chapter 8, "Getting Started with Calculations," **p. 197**, and Chapter 15, "Advanced Calculation Techniques," **p. 351**.



If your calculation formula looks correct but FileMaker is returning an odd result or ?, see "Mismatched Calculation Results" in the "Troubleshooting" section at the end of this chapter.

TIP

You can use calculations to create calculation fields with data derived from other fields or constants. Calculation can also format data, just as fields in layouts can be used to format data.

In general, good database design separates the presentation of data from the content of data, and layouts are the primary tools to be used to format data. However, with FileMaker, the situation is now not so clear. Because you can access FileMaker Pro databases over the Web, with ODBC, and from remote copies of FileMaker Pro that use their own layouts, you might want to consider formatting data with calculations, rather than layouts. Calculation fields that round a number to two decimal places or that perform automatic formatting of dates and so forth produce formatted results visible to all potential users of the FileMaker Pro database, not just those using a layout in the database itself.

SUMMARY

Summary fields allow you to evaluate information across a found set of records. Sum, Average, Max, Min, and Count are among the summaries you can establish. Don't forget that they apply to found sets: Change your found set, and the result changes.

- Summary fields can be placed in subsummaries where they summarize data for a specific subset (perhaps individual clients or dates). FileMaker Pro takes care of summarizing only the appropriate data as you will see in Chapter 4, "Working with Layouts" **p. 103**.

For example, say you have a table called Transaction, which contains Transaction_Date and Transaction_Amount fields. You can then define and place a summary field on a layout to total the Transaction_Amount field. The summary field adds the values of the Transaction_Amount fields for the currently active set of records. If you perform a find, by date, on 10/1/2008–10/31/2008, your found set will be all the transactions for the month of October, and the summary field will show just the aggregate monthly transaction amount. Perform a different find request and your total changes, reflecting the aggregate of the new found set. Table 3.1 contains a list of summary field functions.

TABLE 3.1 SUMMARY FIELD FUNCTIONS

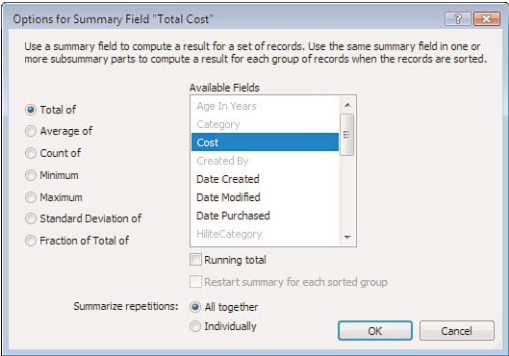
Function	Summary Behavior
Total of	<p>Adds values from the specified field in your found set. Think of it as a subtotal or grand total from a column of numbers.</p> <p>You can also enable the option to display a running total for your record set. This shows a running tally of your total if you place the summary field in the body area of a list.</p>
Average of	<p>Averages the values from the specified field in your found set.</p> <p>The weighted average option enables you to specify a second field to act as a weight factor for calculating the average.</p> <p>The field you choose must be a number or a calculation with a number result.</p>

continues

TABLE 3.1 CONTINUED	
Function	Summary Behavior
Count of	Counts the number of records in your found set that have data in the specified field. For example, if 18 of the 20 current found records have data, your summary field displays 18. A running count functions similarly to a running total: It displays the incremented count of each record in your record set.
Minimum	Returns the lowest number, date, time, or timestamp in a given found set from the referenced field.
Maximum	Returns the highest number, date, time, or timestamp in a given found set from the referenced field.
Standard Deviation of	Determines how widely the values in the referenced field differ. The function returns the standard deviation from the mean of the values in your found set. The standard deviation formula is $n-1$ weighted, following the normal standard deviation. Standard deviation comes in two flavors; to perform a biased or $n-0$ evaluation, select the By Population option.
Fraction of Total of	Returns the ratio of a total for which a given record (or set of records, when the field is placed in a subsummary part) is responsible. For example, you can track what percentage of sales is attributable to a given person. The subtotaled option enables you to specify a second field by which to group your data.

When you create a summary field, the Options for Summary Fields dialog opens, prompting you to choose the function you want to use and the field for which you want a summary (see Figure 3.5).

Figure 3.5
Summary fields are useful for performing functions across sets of records.



In Browse mode, a summary field evaluates your found set and displays a result when it is actually visible on a layout. For example, if a summary field is below the visible portion of a layout, it displays information only when the user scrolls to that portion of the window. Summary fields evaluate a found set for a given layout whenever you enter Preview mode, which is the logical behavior for printing—the primary use of Preview mode.



In FileMaker Pro 10, subsummary fields are displayed and updated in Browse mode as well as Preview mode.

WORKING WITH FIELD OPTIONS

In addition to establishing fields and assigning data types, you can assign various options to your fields as well. These options range in function from managing auto-entry of default data to validation checks and internal storage settings. They can vary for each field type.

After you name a field and choose its type on the Fields tab of the Manage Database dialog, click Create to save it to your database. You can then opt to apply further behaviors via the Options button on the right. The first set of options is the auto-entry behaviors.

3

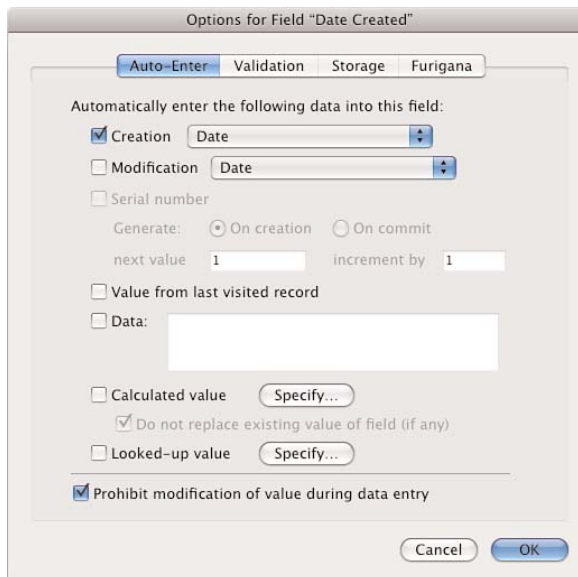
AUTO-ENTRY FIELD OPTIONS

When defining noncalculation fields in FileMaker Pro, you can choose to have data automatically entered into a field as records are created and/or modified. The applications for this can range from assigning default values to fields, to automatically reformatting data, to inserting values from other fields based on certain trigger events.

In some cases you might also want to prevent users from modifying these auto-generated values, such as when tracking a serial ID or applying a date you don't want adjusted afterward (see Figure 3.6).

Figure 3.6

FileMaker's auto-entry options allow you to define rules for automatically populating data into fields in your database.



Based on some trigger event, FileMaker inserts auto-entry data into a field. The most common event is record creation: When a user clicks New Record, data can be prepopulated into the record and be accessible for making changes. Each auto-entry function has its own particular rules for what trigger event applies. In addition to new record creation, other trigger events include record modification and modification of a particular field. We cover both cases in the sections that follow.

CREATION AND MODIFICATION

The first two options on the Auto-Enter tab deal with tracking and applying certain values as a record is committed to your database. They behave essentially the same way, with Creation values being applied the first time a record is committed, and Modification values applied thereafter as it is subsequently modified (committed again).

Values that can be automatically entered include the current date, current time, current timestamp, current username (from the General tab of the Preferences dialog under the Edit menu), and current account name (the one entered by the user when logging in to the database).

CAUTION

Both the name and account name can be problematic because users can change them. Knowing how your solution will be used can help you to decide what value to use. The name value is the name of the computer user—obviously not a good choice if your database will be used in a public library. The FileMaker environment typically controls account names, so they can be a better choice as long as people do not share them.

NOTE

If you do not change any of the account settings of a new file, FileMaker establishes two default accounts for you: Guest and Admin. Both begin with full access to the database.

SERIAL NUMBER

Using the Serial Number option allows you to auto-enter a number that increments every time a new record is added to the table. Often this number uniquely identifies individual records in a table. The value can be generated either when the record is created or when it is committed. The difference is subtle: In the case of incrementing on creation, your number increments even if a user reverts and effectively cancels a record's creation. The next record will then have skipped a number in your sequence. This doesn't have much of an effect on your database unless your business requires strict tracking of each serial number, even those voided. In those cases, choosing On Commit helps avoid spaces in the sequence.

It is possible to include text characters in addition to a number as the starting value if you want. This enables you to create serial numbers that look something like "a1, a2, a3, a4..." Only the rightmost numeric portion of the value is incremented; the text portion remains unchanged. If you do this, you will want to use a Text field to allow for the alphanumeric combination.

One of the common uses of auto-entry options is in establishing serialized key values or IDs. This is a vital element of your database structure when you're working with more than one table, but we encourage you to adopt some best practices regardless of how complex or simple your plans.

For every table in your database, the first field you should create is a primary key or ID field. These IDs uniquely identify each record in your database. You could go about having the system establish unique IDs automatically in several ways; our recommendation in most cases is to use a serial number set to increment automatically.

We can't stress this practice strongly enough. If you ever want to tackle relational data structures, these serial IDs are a vital element in doing so. Further, if you ever export your data to another system or need to interact with other databases, having a key field that uniquely identifies each record in your database guards against confusion or even possible loss of data integrity. To create a serial key field, use the following steps:

1. Define a number field. It is generally advisable to use number-based serial keys, but it is possible to use text as well; the important point is to make certain your keys are unique and users cannot modify them.
2. Go into the Options for that field and select the Serial Number option.
3. Click the Prohibit Modification of Value During Data Entry option at the bottom of the dialog. This is an important step: If you establish unique identifiers that your users can override, you're risking the chance that they'll introduce duplicate IDs.

If you need an ID field for a business purpose (SKUs, student IDs, employee IDs from your organization, and so on), we recommend that you create separate fields for such cases. Generally, users should never need to access this serialized ID field, but you can opt to put it on a layout and allow entry in Find mode so that they can search if they choose.

→ For a full discussion of the use of keys (or *match fields*), **see** the discussion in "Working with Keys and Match Fields," **p. 164**.

VALUE FROM LAST VISITED RECORD

Used most often as a way to speed data entry when information repeats often for groups of records, the Value from Last Visited Record function copies the value from a prior record into a given new record. Bear in mind that *Visited* means the last record in which you entered data. If you enter data in a record and then view a second record without clicking into and activating a field, a new record obtains its value from the data in the first, edited record.

DATA

In the Data field, you can specify literal text for auto-entry. This is frequently used to set default states for field entry. For instance, in an Invoice table, you might have a text field called Status where you want to enter Not Paid as a default. As a regular text field, the value is still fully modifiable by a user.

CALCULATED VALUE

In addition to establishing a field as a calculation field, where a defined formula determines its value, it is possible to use the Calculated Value option to insert the result of a calculation into a field of another type, including a container field, by using an auto-entry option.

Furthermore, if you uncheck the Do Not Replace Existing Value for Field (If Any) option, the result of the calculation formula is entered into the field, overriding any existing value, anytime a field referenced by the calculation changes.

Put differently, any field referenced in your calculation statement acts as a trigger: Anytime that referenced field updates, the calculation retriggers and puts its result back into the auto-entry field.

→ To learn more about advanced calculation functions, including custom functions, see Chapter 15, “Advanced Calculation Techniques,” p. 351.

LOOKED-UP VALUE

The Looked-Up Value auto-entry option copies a value from a record in a related table into a field in the current table. (If there are multiple related records, the value from the first record will be copied; this means that you might want to think twice about using looked-up values for relations in which there might be more than one related record.) Anytime the field controlling your association to the related record changes, FileMaker Pro updates the value in the lookup field. For example, if a user enters a postal code into a given record, it's possible you could have another table auto-populate your city and state fields with the appropriate information.

When a user enters a postal code in the record, the City and State fields trigger to pull values from the ZipCodes table. An important fact to keep in mind is that FileMaker *copies* the values from the ZipCodes table. If the source data changes or is deleted, this record remains unmodified until it is retriggered by someone editing the Zip Code field again.

Take special note that lookup auto-entry functions work just as all auto-entry functions do: They copy or insert information into a field. You are not displaying related information, nor are you controlling content by calculation. Thus, lookup values are not live links to related data. If you were to delete the records in the ZipCodes table in the preceding example, all your people records would remain untouched, preserving your city and state data.

Understanding this distinction is important, especially as we get into indexing later in this chapter. Consider an example for product prices: If you were to build an Orders database that tracks the prices of products, you would want to store the price of each Order line item or product within the order itself. That way if your prices change, your historical orders preserve their original prices. To see how to create a lookup field, refer to Figure 3.7.

Figure 3.7

Often you'll want only exact matches, but in some cases you can use the closest value based on a comparison of the trigger values in your related table.

Lookup for Field "State"

Use a lookup to copy a value from a field in another table based on the relationships defined between these tables:

Starting with table: **DataTable**

Lookup from related table: **PostalCodes**

(When a new entry is made in the field "State", this lookup will copy the value from the first matching related record in the table "PostalCodes".)

Copy value from field:

- ☐ ::PostalCode
- ☐ ::City
- ☐ ::State

If no exact match, then:

- ☒ do not copy
- ☐ copy next lower value
- ☐ copy next higher value
- ☐ use

☒ Don't copy contents if empty

Cancel OK

Remember that any time your match field changes, your lookup refreshes. In this case, the auto-entry function does not act on record creation, but rather on committing.

When you're performing a lookup, it is possible to work with near matches in addition to exact matches. In the case of the postal codes example, obviously you would want only an exact match or you might end up with incorrect data. In a different case, however, you need not be so strict. Consider a scheduling system that automatically finds the closest available appointment: Enter a target date into a field, and the lookup function could return the closest match. Another application might be a parts database with units of measurement. You may not be able to find a .78" wrench, but a .75" might work. This sort of requirement is easy to meet by using the Copy Next Lower Value setting (or its higher value companion).

How you set up your matching field values is important here. It's easy to compare numbers and come up with the next closest value. If your matching field is text, FileMaker Pro uses ASCII value rules to compare and determine order.

➔ For further discussion of lookups, **see** Chapter 6, "Working with Multiple Tables," p. 159.

HOUSEKEEPING CREATION AND MODIFICATION FIELDS

As a best practice, we also recommend that you create another set of fields in all tables that help track changes. Create a timestamp field and in the Auto-Enter options, choose Creation Timestamp. Define another timestamp field for Modification Timestamp, and text fields for Creation and Modification Account Names.

These four fields tell you exactly when a record was created or modified and by whom (assuming that you assign an account to each individual person using your database). If you ever need to identify problem records for a given day range, time, or account, these fields allow you to do this. We strongly recommend that you add them every time you create a new table. The only downside to following this practice is that additional storage space is required for this data; in this version of FileMaker Pro, this is unlikely to be a concern.

TIP

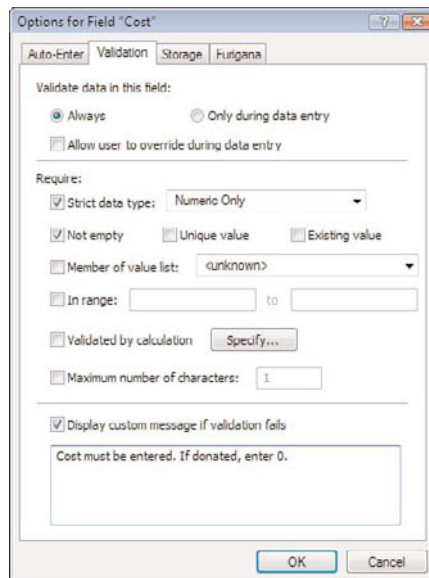
Using FileMaker Pro's capability to import tables allows you to create a boilerplate new table, complete with a primary key serial ID, four housekeeping fields, and whatever other standard fields you want to define. Whenever you need to add a table to your database, import from the boilerplate rather than having to re-create these standard fields. If you are using FileMaker Pro Advanced, you can use the copy and paste commands for fields or a table containing these fields.

FIELD VALIDATION

Storing correct and complete information is critical for generating accurate reports; establishing proper, expected conditions on which other functions and calculations are performed; and ensuring overall data integrity. Unfortunately, most data applications suffer from a chronic condition of having humans interacting with them; although some humans are worse than others, no one is perfect. We all make mistakes.

As a user enters data into FileMaker Pro, you might opt to apply one or more validation checks to test that a record meets certain conditions before allowing the user to commit it to your system. This task can be as simple as ensuring that a field isn't empty or as complex as making sure that an invoice doesn't contain multiple entries for the same product. To review the various validation options available, see Figure 3.8.

Figure 3.8
You can set validation rules for the database fields.



This example demonstrates a common approach to ensuring proper maintenance of your primary keys. This might be overkill if you've enabled the Prohibit Modification of Value During Data Entry option on the Auto-Enter tab, but on the chance that a developer turns that option off for some reason or that users import records into your database, this is a handy bit of insurance.

- ➔ Importing records can circumvent your carefully designed field validation rules. For a full discussion, **see** Chapter 22, "Importing Data into FileMaker Pro," **p. 493**.

VALIDATION CONDITIONS AND FAILURE

Field validation simply tests whether one or more conditions, as defined in your Validation dialog, are false. If all validation tests are true, FileMaker Pro does not interrupt or prompt the user for action. Figure 3.9 shows an example of what your users might see when validation fails.

Figure 3.9
The OK option appears only if a user has the option to override the validation warning.



In this case, the check box allowing users to override has been left enabled, so they have the option to ignore the warning. When that function is disabled, the field does not allow bad data to be committed, and the system forces users to deal with the problem. They can choose either to revert the field to its previous state or to clear it.

WHEN VALIDATION OCCURS

Validation occurs when users manually enter data into the field being validated; some validations happen the moment the user leaves the field, whereas other validations are deferred until the user commits the record. Remember, however, direct entry is not the only way to get information into a field. You can also import records or use various script steps, such as `Set Field()`.

Simply clicking or tabbing into a field does not trigger validation; a change has to be attempted. Keep in mind that validation does not apply in cases in which users modify other, nonvalidated fields of a given record. A given field's validation check occurs only when data in that specific field changes.

At the top of the Validation tab of the Options dialog (refer to Figure 3.8), notice the Always and Only During Data Entry choices. The latter choice tests for validation conditions only when users modify the field in question. If you enable the Always option, validation occurs during scripts and imports as well as during data entry.

If an import process attempts to write invalid data to a field, FileMaker Pro simply ignores the improper entry. The field remains unchanged and does not import your data. You will see a note in the Import Records Summary dialog listing how many errors FileMaker Pro encountered. If you enable the Only During Data Entry option, FileMaker Pro would insert the improper data into your database.



If you get trapped in a series of validation dialogs, refer to “Validation Traps” in the “Troubleshooting” section at the end of this chapter.

TIP

Some designers make a distinction between validation errors and quality errors. In general, *validation errors* must always be corrected and can never be ignored by users. They are hard and fast rules about the data: no nonnumeric data in a numeric field, no missing data, and so forth.

Quality errors (which FileMaker Pro nevertheless implements through the Validation tab of Options) can be overridden. You can construct a quality edit based on a calculation that compares the entered value to the value from the previous record; a difference of more than a certain margin might result in a flag and require the user to confirm the value. This type of quality checking can catch many keying errors.

STORAGE AND INDEXING

Field storage and indexing options exist on the Storage tab in your Field Options dialog. These options control how FileMaker Pro indexes each field to speed up searches and sorts and form relationships.

GLOBAL STORAGE

A developer can designate a field to have global storage on the Storage tab of the Field Options dialog. Fields with this option are commonly referred to as *global fields*, and collectively they’re usually referred to as *globals*. Global fields exist independently from any specific record in the database and hold one value per user session. Developers often use global fields to establish special relationships or to display unchanging information, such as interface graphics or field labels, across multiple records and layouts.

One vital element to learn is when data is committed and stored for globals: In a single-user environment, any change to a global field is permanent and saved across sessions. In other words, whatever value you last entered into a global will remain the next time you open your database. In the case of a multiuser environment, where a FileMaker Pro solution is hosted on FileMaker Server or via multiuser hosting, global values for each guest default to the value from the last time the database was in single-user mode; any change made to these defaults will be specific only to a given user’s session. Other users continue to see the default values, and after the database session is closed, the database reverts to its original, default state.

NOTE

In the case of globals with values that can change, it is good to initialize them in a startup script. This could mean having pairs of globals. One of them can never change, and the other one can be changed by various users at various times, but you will always reset it in a startup script to the unchanging value. Now that local and global variables are available, it is often the case that they are better suited than global fields for values that might change.

Using globals is a great way to keep track of certain states of your database. For example, you could use a global field to store which row of a portal was last selected. This field could then be used in scripts or calculation formulas.

Another common use of globals is for storing system graphics. Establish a container field, set it for global storage, and paste a favorite company logo, a custom button graphic, or any number of elements that you can then control globally in a field rather than having to paste discrete elements on each and every layout.

Beginning with FileMaker 8, a new feature was created in the form of variables defined within scripts (as well as similar variables defined by using the `Let()` function within calculations). These variables exist only in memory and are not permanent fields that you add to your database schema. In the past, developers had to content themselves with using a slew of global fields; starting in FileMaker 8, the need for global fields has dropped considerably. However, you will still encounter them in legacy databases.

→ To learn more about variables in FileMaker, **see** Chapter 16, “Advanced Scripting Techniques,” **p. 387**.

REPEATING FIELDS

The second section of the Storage tab on the Field Options dialog lets developers allow a field to contain multiple values. Such fields are known as *repeating fields*. On a given layout, the developer can array repetitions either horizontally or vertically, and in scripts can refer to specific repetitions within the field.

Repeating fields can be problematic. They behave just as individual fields might and are really just a shortcut for having to define multiple instances of a given field. It's possible, for example, to have no values in the first and second repetitions, but to have a value in the third. This sounds convenient and makes sense intuitively, but imagine having to write a script that references that field. How do you know which repetition of the field to reference? Unlike an array in other programming languages, you cannot manipulate a repeating field as a whole. You can reference only one specific repetition at a time.

FileMaker 8 extended the usefulness of repeating fields somewhat by allowing the script step `Set Field` to programmatically reference a repeating instance. You can now open a Specify Calculation dialog to point a script to a specific cell within a repeating field. Note that the same is true for setting variables.

Repeating fields do have their place, however. Sometimes a single data value does have several components. An RGB color, for example, has three values: one for red, one for green, and one for blue. Creating an RGBColor field with three repetitions makes a great deal of sense.

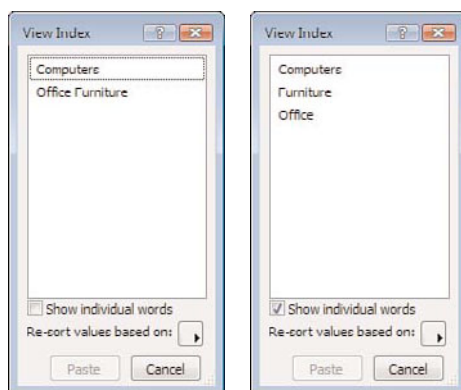
INDEXING

Databases store data, of course, but they are also required to perform functions such as searches and sorts with that data. FileMaker Pro, like many databases, can index some of the data in a file to increase the speed at which it performs some of these functions and to enable it to relate data across tables.

An *index* is somewhat like a database within a database. FileMaker Pro can store, along with a specific value in a given field, a list of all the records in which that exact data is used. This enables FileMaker to recall those records quickly, without having to resort to a linear scan of your file. Aptly named, these indexes work just as a book index works: They facilitate finding all the locations in which a given item is used, without searching page by page through the entire book.

To familiarize yourself with the concept, look at a given field's index. Click into a field and select Insert, From Index. If the field is indexable, and has already been indexed, you see a dialog showing all the discrete values indexed for a given field. Just as when selecting from a value list, you can opt to choose from this list rather than type. As you can see in Figure 3.10, FileMaker Pro can create the index based on data values or individual words.

Figure 3.10
You can view index values using From Index in the Insert menu.

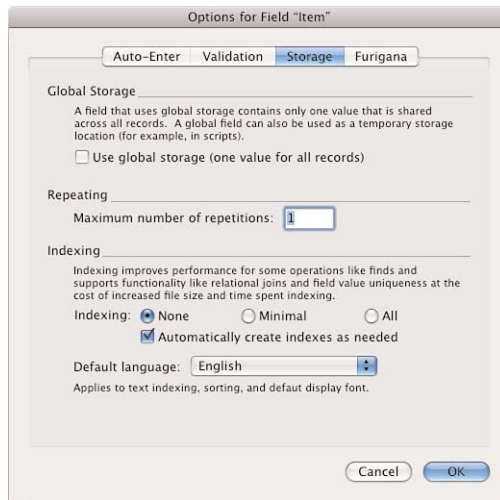


Allowing a user to select from an index is only one of the reasons to use indexes in FileMaker. Indexes enable FileMaker Pro to quickly perform find requests, sort records, and establish relationships.

There are two kinds of indexes in FileMaker: value indexes and word indexes. *Value indexes* apply to all field types, with the exception of container or summary fields. *Word indexes* apply only to text fields and are based on a given language or character set. The difference between the two index types, and when either is specifically enabled, lies in their applications.

FileMaker Pro's default indexing setting (found on the Storage tab of the Field Options dialog, displayed in Figure 3.11) is None, with the check box for Automatically Create Indexes As Needed enabled. Most developers, even the more advanced, should find that this setting serves most of their needs. (The Minimal setting produces a value index for text fields or calculation fields returning text. For all indexable fields, the All setting creates a value index, and for text fields or calculation fields returning text, it also produces a word index.)

Figure 3.11
FileMaker creates either one type of index or both, depending on how users define and use a field.



A database's schema definition establishes value indexes, as a developer defines fields and builds relationships. In addition, value indexes allow for relationship matches and value lists. If a developer creates a serial ID and joins a relationship via such a field, FileMaker Pro creates a value index for the serial ID field.

Unless a developer explicitly sets a field to generate an index, FileMaker Pro creates word indexes as users are interacting with and using a given database. Word indexes are utilized in text fields for find requests; they are created when a user explicitly chooses Insert, From Index. If a user enters data in a find request for a field that lacks a word index, FileMaker Pro enables indexing for that field and builds one (unless it's explicitly unindexed or an unindexable calculation).

At this point you might be wondering what all the fuss is about. Why not index every field in a database and be done with it? The downside to indexes is increased file size and the time it takes FileMaker to maintain the indexes. Creating new records, and deleting, importing, and modifying them, all take more time, in addition to the fact that the indexes themselves take up more file space.

Notice that FileMaker doesn't allow you to explicitly control word and value indices. Value indices are possible for all field types; word indices apply only to text fields. The Minimal setting is an available option only for text fields, and when you see it marked, it indicates that at least one of the two indices exists for the field. There's no straightforward way of determining which index exists. If you explicitly set the field to Minimal, FileMaker creates, on demand, either of the two indices based on how the field is used. When a user creates a find request including that field, FileMaker creates a word index; if a developer uses the field in a relationship, FileMaker creates a value index.

Only a subset of the fields in your database will ever need to be indexed, and FileMaker's "on demand" approach makes things simple for developers. In general, it's best if a field is indexed only when necessary.

- ➔ To explore the vagaries of storage and indexing considerations for calculation fields, **see** “Options,” **p. 204**.

An important point to remember is that some fields are not indexable. This means that they will be slow when used in sorts and find requests, but, most important, you cannot use them to establish relationships. A field is unindexable if it is a calculation based on a related field, a summary field, or a global field, or if it references another unindexed, unstored calculation field.

You can also explicitly make a field unindexable by turning indexing options to None and unchecking the Automatically Create Indexes As Needed setting. In the case of a calculation field, an additional radio button option is available: Do Not Store Calculation Results—Recalculate When Needed. These settings are important to remember; they allow you to force FileMaker to reevaluate and display dynamic information. The `Get (CurrentDate)` function, for example, displays the current date if you have indexing turned off but displays whatever date was last stored with the record if you leave indexing (and storage) turned on.

3

Furigana

The fourth tab in the Field Options dialog is one that many English-speaking developers will have trouble properly pronouncing, let alone using. Because of the adoption of Unicode support in FileMaker Pro 7, it is now possible to offer Asian-language double-byte language support. As a result, you can now manage Japanese.

Japanese is written using a combination of kanji, complex glyphs borrowed from Chinese that represent complete concepts, and hiragana, a simpler alphabet that represents the phonetic syllables of the language. Furigana is a smaller version of hiragana that acts as a cheat sheet for readers who aren't familiar with a kanji character's reading. The Furigana feature in FileMaker makes it possible to render a kanji-based block of text into its phonetic hiragana equivalent—quite useful when you don't know how to read one of the more than 20,000 kanji characters.

TROUBLESHOOTING



MISMATCHED DATA TYPES

My data isn't sorting properly. Where should I look first to diagnose the problem?

One of the most common bugs you'll run into in FileMaker Pro is confusion stemming from mismatched data types. If your users are entering text data into a field you have defined as numeric, you're bound to get unexpected results, and sorting will be unpredictable. Check your field types when your data appears to be misbehaving.

MISMATCHED CALCULATION RESULTS

One of my date calculations looks like an integer. What's going on?

Some of the more subtle extensions of the data type problem are calculation fields. Note that their result is both the determination of their formula and a data type that you set at the bottom

of the Specify Calculation dialog. If you're working with dates and return a number, for example, you'll get an entirely valid calculation that will look nothing like "12/25/2003."

PROBLEMATIC FIELD NAMES

My web programmers are complaining about my field names in FileMaker Pro and that I keep changing them. What should I consider when naming fields?

Some other systems are not as flexible as FileMaker Pro; this is especially true for URLs and the Web. Spend some time with Chapters 24, 25, and 26 if you ever plan to publish your database to the Web. FileMaker Pro breeds a certain freedom when it comes to changing field names as the need arises, but you'll send your XSLT programmer into fits every time you do.

Also be sure to check the restrictions of various SQL databases in your organization. If you need to interoperate with them, your field names might have to conform to stricter naming standards.

You'll be safe if you never use spaces or special characters and start each field with a letter of the alphabet or an underscore.

VALIDATION TRAPS

My field validation seems to have gone haywire. I defined a field that now simply throws up one error message after another. What's the problem?

At the end of the day, field validation is only a helpful bank of sandbags against the storm of human interaction your database will suffer. And as in all aspects of your database, the first and worst human in the mix is the developer. Just as with any programming logic, carefully test your validation conditions. FileMaker Pro can't totally prevent you from illogically conflicting restrictions. For example, if you set a field to be unique and nonempty but also prohibit modification in the auto-entry options, the first record you create will trap your system in an irresolvable conflict.

It's a good idea to leave the Allow User to Override During Data Entry option enabled while you're building a solution and turn it off only after you have completely tested the field in question.

RE-CREATING INDEXES

I am getting find errors returned for valid requests. What has happened? What should I do?

These errors can be a symptom of a corrupted index. In Manage Databases, go to the Fields tab and select Options, then the Storage tab. This is the place where you manage indexes. Note the settings; then click the None check box and turn off Automatically Create Indexes as Needed. Close the various dialogs until you are back in FileMaker itself. If you want to be absolutely safe, quit FileMaker, restart it, and then reopen the database. You will then have no indexes on the field in question. Go back to Manage Databases, through the Fields tab, Options button, and the Storage tab. Turn indexing back on using the settings that you noted. The index is re-created and should be correct.

FILEMAKER EXTRA: INDEXING IN FILEMAKER

One of the more significant changes beginning in FileMaker 7 revolves around indexing. In prior versions, indexing was restricted to 60 characters total, broken into blocks of up to 20-character words. Relationships had to be built around match fields or keys that were relatively short and generally nondescriptive. This fact is one reason we generally advocate using simple serial numbers for indexing purposes. It's rare that you'd need more than 20 digits to serialize the records in a data table.

FileMaker 10 can index words up to approximately 100 characters. It can index text fields to a total of 800 characters, and numbers up to 400 digits. The limits to indexing have been effectively removed.

What this means to developers is that you can now use far more complex concatenated key combinations (ironically, there will be less of that in FileMaker 10, given that data can be related across multiple tables), use longer alphanumeric keys, or, as suggested earlier, introduce descriptive elements to keys.

TIP

This bit of history is particularly useful if you are working with older FileMaker solutions. Even though you are looking at a database in FileMaker 10, it may have had its roots in a much earlier version. Workarounds with indexes often survive, leaving mysterious remnants for you to try to figure out.

INDEX

Symbols

- & (ampersand) operators, 202
- * (asterisk) wildcards, 525
- + (plus sign) operators, 202
- (hyphens), layout names, 115

A

- abbreviations, decoding, 355
- Abs functions, 218
- absolute paths, 458
- abstraction (scripts), 237
- access logs, IWP, 526
- access privileges
 - matrices, 457
 - troubleshooting, 321
- Access via FileMaker Mobile extended privilege, 315
- Access via FileMaker Network extended privilege, 314
- Access via Instant Web Publishing extended privilege, 314
- Access via ODBC/JDBC extended privilege, 314
- Access via XML Web Publishing extended privilege, 315
- Access via XSLT Web Publishing extended privilege, 315
- accessing
 - custom functions, 375
 - plug-in names/version numbers, 630-631
- Accounts and Privileges menu, 529
- Accounts tab (Manage Accounts & Privileges dialog), user accounts
 - privilege sets, reviewing, 305
 - security, 303
- ACID tests, 288-289
- activation process, 25-26
- Add Account script step, 322
- Add Table Occurrence icon (Relationships Graph), 191
- adding
 - external ODBC data sources to Relationships Graph, 486
 - fields to layouts, 135-136
 - files to runtime applications, 591
 - merge fields to layouts, 139
 - pictures to layouts, 123
 - related records to master records, 169
 - search paths to file references, 189
 - tables to multitable systems, 161
- Admin accounts
 - access, removing via Developer Utilities (FileMaker Developer), 593
 - passwords, troubleshooting, 321
 - user-level internal security, 304
- Admin Console
 - Administration section, 618-619
 - scheduling feature, 620-621
 - statistics, monitoring, 622
 - Configuration section, 623
 - database cache, RAM allocation, 625
 - web publishing settings, 625
 - CWP URL requests, troubleshooting, 573
 - FileMaker Server access, troubleshooting, 573
 - launching, 617
 - WPE access, troubleshooting, 573
- Administration section (Admin Console), 618-619
 - scheduling feature, 620-621
 - statistics, monitoring, 622
- Advanced Web Publishing Options dialog (IWP Database Homepage), 524
- aggregate functions, 221-222
- Align command (Arrange menu), 132
- aligning layout objects, 132
- All Modifiable setting (Edit Privileges dialog), 310

All No Access setting (Edit Privileges dialog), 310

All View Only setting (Edit Privileges dialog), 310

Allow Creation of Records in This Table via This Relationship check box (Edit Relationship dialog), 168

Allow Exporting dialog (Edit Privileges dialog), 313

Allow Printing dialog (Edit Privileges dialog), 313

Allow User Abort scripts, 240

Allow User Abort setting (IWP), 534

Allow User to Modify Their Own Password option (Edit Privileges dialog), 313

alpha conversion, 454

Also Reduce the Size of the Enclosing Part option (Set Sliding/Printing dialog), 278

Alternate Background Fill option (Part Definition dialog), 119

alternating row color in reports, 274

Always During Data Entry option (Validation dialog), 95

Always Lock Layout tools option (Layout mode), 123

ampersand (&) operators, 202

application flow, IWP, 538
 creating file links, 541
 explicit record commits, 539
 hiding Status toolbar, 540
 portals, 540
 session management, 538

application logs, IWP, 526

applications, ODBC, 472-473

Arrange menu (Layout mode)

 Align command, 132
 Bring Forward command, 131
 Bring to Front command, 131
 Distribute command, 132
 Move to Back option, Tab Control objects, 134
 Object Grids command, 127
 Resize To alignment tools, 127
 Resize To command, 132
 Send Backward command, 131
 Send to Back command, 131

array functions, 364

 GetValue functions, 366
 LeftValue functions, 365, 383
 MiddleValues functions, 365, 383
 RightValues functions, 366, 383
 stepping through, 366
 usage examples, 366
 ValueCount functions, 365, 383

arrays

 elements, navigating, 366
 text arrays, passing multi-valued script parameters, 390

Asian-language double-byte language support, 100

asterisk (*) wildcards, 525

atomicity (ACID tests), 289

attributes (database design), 144-145

 join entities, 154
 versus entities, 146-147

audit trails, troubleshooting
 auto-entry techniques, 295

authenticating

 external authentication, 319-320
 IWP accounts/privileges, 531
 users, 318

Auto Update feature (FileMaker Server), 626

auto-entry options (fields), 89-90

 Calculated Value option, 92
 calculation formulas, 200
 Creation option, 90
 Data option, 91
 Looked-Up Value option, 92-93
 Modification option, 90
 Serial Number option, 90-91
 Value from Last Visited Record option, 91

automatic logins, user-level internal security, 304

Automatically Create Indexes as Needed check box (Indexing option), 206

AutoUpdate plug-in

 directories
 building, 626
 folder naming structures, 627
 enabling, 628
 FMSUAC_FindPlugin function, 628-630
 FMSUAC_UpdatePlugin function
 downloading plug-ins, 631-632
 error codes, 629
 FMSUAC_Version function, 628

Available Menu Commands option (Edit Privileges dialog), 313

B

background colors, layout parts, 118

backups
open files, 603
scheduling, 621, 634

Bento, 508

The Bento Book: Beautify and Simplicity in Digital Organization, 508

beta conversion, 455

bidirectional relationships, 181

binary search technique (debugging scripts), 450

bind keys, adding/replacing files in runtime applications, 591

Bindkey option (Solution Options), 591

Blank Layouts, 106

Boolean formulas, 199

Boolean tests, Choose functions, 355

bound runtime applications, adding/replacing files, 591

branching, conditional script steps, 247-248

break field, 265

breakpoints (scripts), 450

Bring Forward command (Arrange menu), 131

Bring to Front command (Arrange menu), 131

Browse mode, 32
dependencies, table context, 441
list reports, 262

Browse mode (IWP), 542

building

AutoUpdate directory, 626

buttons, 253

custom function libraries, 384-385

field matches, 178

HTML-formatted search result stylesheets, 565-566

many-to-many relationship structures, 172

multiple find requests, 556

one-to-many relationships, adding tables to multitable systems, 161

relationships with global values, 187

script libraries, 255

website with XSLT Site Assistant, 564-565

windows, troubleshooting, 424

Button Definition dialog, accessing, 136

Button Setup dialog, Current Script option, 419

buttons

building, 252-253

fields as, 136

FileMaker Pro buttons, 34-35

C

calculated replaces, 200, 437

Calculated Value option (field auto-entry options), 92

Calculation Context section (Specify Calculation dialog), 207-209

calculation fields, 49, 85-86, 197. *See also* calculations

data types, 204, 226

global storage, 205

indexing, 206-207

number of repetitions, 204

Calculation Result Is feature (Specify Calculation dialog), 86

calculations, 403

comments, 430

debugging, 434

formulas, 197

auto-entry options (fields), 200

calculated replaces (fields), 200

context, specifying, 207-209

debugging, 353

field names, troubleshooting, 225-226

field validation, 200

record-level security, 200

script steps, 199

writing, 198-202

functions

Exact function, 213

Length function, 212

PatternCount function, 212

Position function, 213

Substitute function, 214

Trim function, 213-214

WordCount function, 213

nontext, troubleshooting text formatting functions, 382

slowness (performance), 437

table context, 440

writing tips, 226-228

calendars, Drop-Down Calendar option (Field/Control Setup dialog), 136

camel case, 430

Cancel buttons, designing dialog windows, 418

candidate entities, 147

capturing

errors, troubleshooting, 295

record locking in scripts, 292

cardinality, Relationships Graph, 164

carriage return transformations (exporting data), 514

carriage returns in fields, 137

Cartesian product relationships, 187

case altering functions, 214

Case statements, conditional functions, 221

Ceiling functions, 217

centering layout objects, 132

change Password script step, 323

character transformations (exporting data)

carriage returns, 514

CSV format, 515

DBF file format, 516

DIF, 516

Excel file format, 515

FileMaker Pro file

format, 515

HTML Table format, 515

Merge format, 515

repeating fields, 514

Tab character, 514

tab-separated text

format, 515

WKS file format, 516

XML file format, 515

Check box set formats (fields), 46

Checkbox Set option (Field/Control Setup dialog), 136

child entities, 150

child files, related parent data, 171

child records, 56

Choose functions, 354-355

Closing Splash Screen option (Solution Options), 592

closing/opening files, troubleshooting, 322

code, writing

comments, 430-431

naming conventions

database files, 428

fields, 429

functions, 430

layouts, 429

parameters, 430

collection lists

employee tables, example

of, 12

flat files, 14

phone directory flat files,

example of, 12

relational databases,

example of, 12-13

color

background colors, layout

parts, 118

reports, alternating row

color, 274

Columnar List/Report layouts, 106

columns

multi-column layouts,

113-114

sortable headers, 276

commands (custom menu components), 341

Comment dialog (field definition dialog), 430

Comment text box (Manage Database dialog), 81

comments

fields, adding to, 81

proactive troubleshooting,

430-431

scripts, 235

viewing, 430

Commit button (IWP controls), 539

Commit Record/Request script step, 292

Commit Record/Request step (IWP), 535

committed data versus created data, 290

committed records, 44

comparison operators, -find command, 555

completing scripts, Allow User Abort scripts, 240

concatenation, 201, 212

concurrency, 288-289

conditional formatting, 348

conditional functions, 220-221

conditional privileges, 310

conditional script steps, 247-248

loops, 250

troubleshooting, 254

conditional tests, Choose functions, 354

Configuration section (Admin Console), 623

database cache, RAM

allocation, 625

web publishing settings, 625

configuring

custom functions, 375

databases for CWP, 548-549

fields, 137

FileMaker databases for

ODBC, 473-474

FileMaker Server for

automatically updating

plug-ins, 627-628

IWP, FileMaker Pro 8

configuration, 526

plug-ins, 599

portals, 166

Web Viewer, 325, 328

connecting to FileMaker Server with PHP Site Assistant, 580

connectivity, troubleshooting server connections, 438-439

consistency (ACID tests), 289

Consistency Checker (FileMaker Server), 603

- constrain requests, 63**
- Constrain Found Sets, 63**
- constructors,**
 - Time/Date/TimeStamp functions, 219**
- container fields, 49, 84-85**
 - Export Field Contents option, 51
 - exporting data, 518
 - Import option, 50
 - Insert option, 50
 - IWP restrictions, 537-538
 - OLE support, 50
 - Paste option, 50
 - preparing for Web Publishing Engine, 578
 - saving/retrieving information in, 50
- context, troubleshooting**
 - layout dependencies, 439
 - record pointers, 442
 - table context, 439-441
- Continue buttons, designing dialog windows, 418**
- controls, adding to Web Viewer, 330-331**
- converted passwords, troubleshooting, 321**
- converting files from previous Filemaker versions, 454-455**
 - multifile relational solutions, 456-457
 - post-conversion tasks, 460
 - formula-related issues, 465-468
 - layout-related issues, 468
 - relationships, organizing, 462-463
 - script-related issues, 463-465
 - security, 461-462
 - preconversion tasks
 - file references, fixing, 458
 - files, examining, 460
 - solutions, documenting, 457
 - single-file solutions, 455-456
 - troubleshooting, 468-469
- converting web-enabled databases, 469-470**
- Copy command (Preview mode), 441**
- Copy script steps, 242**
- copying/pasting**
 - fields, troubleshooting, 141
 - layouts, 107, 125
 - open files, 603
- corrupt files**
 - preventing, 443
 - recovering, 445-446
- Count functions, 222**
- Create setting (Custom Privileges dialog), 309**
- created data versus committed data, 290**
- creating**
 - fields, 82
 - layouts, 106-107
 - tables, 78
- Creation Account Names fields, 93**
- Creation option (field auto-entry options), 90**
- Creation Timestamp fields, 93**
- cross-platform layouts, building, 142**
- cross-product relationships, building, 187**
- CrossProduct functions, 379-380**
- crosstalk, multiple files, 189**
- CSV (comma-separated value) format, 515**
- Current Script option (Button Setup dialog), 419**
- custom development software, 11**
- custom dialogs, scripts, 250-251**
- custom functions, 370**
 - access, restricting, 375
 - automatic updates, 373
 - availability, configuring, 375
 - building, 373-375
 - comments, proactive troubleshooting, 430
 - complex formulas, simplifying, 371-372
 - CrossProduct functions, 379-380
 - Hypotenuse functions, 376
 - libraries, building, 384-385
 - naming conventions, 374, 430
 - NthRoot functions, 376
 - PadCharacters functions, 378
 - parameters, naming, 430
 - Quarter functions, 376
 - recursive functions, 373
 - refactoring, 372
 - RepeatText functions, 377
 - system constants, defining, 372
 - TrimChar functions, 379
 - WeekEndingFriday functions, 376
- Custom Image option (Solution Options), 592**
- Custom Menus feature**
 - scripts, error handling, 432
 - security, 301
- Custom Menus tab (Manage Custom Menus dialog), 342**
- Custom Privileges dialog (Manage Accounts & Privileges dialog), 308**
 - Create setting, 309
 - Delete setting, 309
 - Edit setting, 309
 - Field Access setting, 309
 - View setting, 309
- Custom Privileges setting (Edit Privileges dialog), 310**
- Custom Script Privileges dialog (Edit Privileges dialog), 312**

Custom Value List Privileges dialog (Edit Privileges dialog), 311**customizing**

- extended privileges, user-level internal security, 315-316
- privilege sets, 308-309
- solutions via Developer Utilities (FileMaker Developer), 594

Cut script steps, 242**CWP (Custom Web Publishing), 545**

- commands, 569
- databases, configuring for, 548-549
- operators, 555
- PHP, 576
 - extended privileges, configuring, 576
 - layouts, reviewing, 577
 - placing files on Web server, 578
 - preparing FileMaker Server for, 577
- selecting technology, 575
- session management, 572-573
- stylesheets, sharing data between, 567-568
- triggers, 406
- URL

- exact match searches, 554-555
- multiple criteria searches, 556
- multiple find requests, 556
- numerical comparison searches, 555
- parameters, 570-571
- requests,
 - troubleshooting, 573
- specific record searches, 554
- specifying search result sort order, 557
- table searches, 553

- versus IWP, 546-547
- WPE access,
 - troubleshooting, 573
- XML publishing, 549-553, 557
- XSLT
 - building HTML-formatted search result stylesheets, 565-566
 - embedding query parameters in stylesheets, 566-567
 - query strings, 566
 - stylesheet placement, 558
 - URL formats, 566
 - XSL extensions, 572
 - XSLT processors, 571

D**damaged files, recovering, 443-446****data**

- editing, audit trails, 295
- fields, inserting into, 72-73
- globally replacing, 71-72
- integrity, troubleshooting, 70
- lost data, troubleshooting, 70
- multitiered sorting, 73
- XML publishing, 549-553, 557

Data Access and Design dialog (Edit Privileges dialog), 312**data configuration scripts, 241**

- Copy script steps, 242
- Cut script steps, 242
- Go to Field script steps, 242
- Paste script steps, 242
- Set Field script steps, 242-243

data control scripts, 241

- Copy script steps, 242
- Cut script steps, 242
- Go to Field script steps, 242

- Paste script steps, 242
- Set Field script steps, 242-243

data entry fields, carriage returns, 137**data imports, troubleshooting, 508-509. *See also* data, exporting****Data option (field auto-entry options), 91****data source names, managing ODBC, 477****data sources, Bento, 508****data transfer, SSL data transfer, 604****Data Type option (Specify Calculation dialog), 204****data validation**

- FileMaker Pro, 52
- override privileges, security, 313

Data Viewer (Tools menu), 451**data, exporting**

- character
 - transformations, 514
- container fields, 518
- fields, selecting, 512-514
- file formats
 - choosing, 512
 - CSV format, 515
 - DBF file format, 516
 - DIF, 516
 - Excel file format, 515
 - FileMaker Pro File format, 515
 - HTML Table format, 515
 - Merge format, 515
 - tab-separated text format, 515
 - WKS file format, 516
 - XML file format, 515
- file privileges, 513
- fixed-width formats,
 - padding data, 517-518
- formatting, 516

- from all records, 514
- grouped data, 517
- large fields, 518
- reasons for, 511
- related fields, 516
- scripts, 519
- source tables, choosing, 512

database design. *See also*

databases

- as iterative process, 148
- attributes, 144-147
- entities, 144-145
 - keys, 151-152
 - relationships, 148-152
 - versus attributes, 146-147
- ERD, 144
 - attributes/entities, 147
 - many-to-many relationships, 154
 - notation, 148-149
 - one-to-one relationships, 149

database extensions, runtime applications, 591

Database Homepage (IWP), 530

DatabaseNames functions, 225

databases, 29

- building, Manage Database dialog, 75-76
- concurrency, 288
 - ACID tests, 288-289
 - committed data versus created data, 290
 - multi-context locking, 293
 - multi-window locking, 292-293
 - record locking, 291-292
 - script logs, 290
 - serial IDs, 290
- CWP, configuring for, 548-549
- data validation, 52

deploying

- FileMaker Server
 - Advanced hosting, 17
 - Kiosk mode (FileMaker Pro Advanced), 17
 - peer-to-peer hosting, 16
 - server hosting, 17
 - single-user
 - deployment, 16
 - single-user runtime deployments (FileMaker Pro Advanced), 17
- directory structures, FileMaker Server maintenance, 634
- fields, indexing, 97-100
- FileMaker Pro, opening in, 40
 - local files, 40
 - remote files, 41
- files, naming, 428
- LAN, opening on, 42
- navigating, 73
- ODBC, configuring, 473-474
- sessions, tracking, 288
- software
 - advantages of, 9
 - as custom development software, 11
 - as off-the-shelf software, 10
 - functions of, 9-13
 - team development, 295-296
 - transactions, 288

date fields, 49, 83

Date functions, 218-220

-db URL parameters, 570

DBF file format (exporting data), 516

-dbnames command, 569

DDR (Database Design Reports), 446

- creating, 447
- HTML, 447
- Word (MS) documents, formatting as, 446

deactivating Filemaker Pro, 26

Debug Scripts feature (Data Viewer), 451

debugging

- calculation formulas, 353
- calculations, 434
- processes, script logs, 290
- scripts, 448, 450
 - inspecting values, 451
 - looping scripts, 451
 - placing breakpoints, 450

declarative

programming, 471

decoding abbreviations, 355

dedicated find layouts, 423-424

defining

- field behaviors, 137
- layout parts, 118-119
- tooltips, 140

Delete Account script step, 323

-delete command, 569

Delete setting (Custom Privileges dialog), 309

deleting

- parts of layouts, 118
- portal rows, 56
- records from FileMaker Pro, 44
- restricted deletes, 176
- tables, 79

deploying

databases

- FileMaker Server
 - Advanced hosting, 17
 - Kiosk mode (FileMaker Pro Advanced), 17
 - peer-to-peer hosting, 16
 - server hosting, 17
 - single-user
 - deployment, 16
 - single-user runtime deployments (FileMaker Pro Advanced), 17

FileMaker Server,
611-612, 616
plug-ins via FileMaker
Server, 599
runtime solutions, 589

Design functions, 224-225

designing

databases
as iterative process, 148
attributes, 144-147
entities, 144-147
entity keys, 151-152
entity relationships, 148-152
ERD, 144, 147-149
reports
alternating row
color, 274
sortable column
headers, 276
user interfaces, 349-350

developer fields, 80, 429

Developer Utilities

(FileMaker Developer)

Admin access,
removing, 593
error logs, 594-595
files, renaming, 587-588
Kiosk mode, enabling,
595-596
Kiosk mode, securing, 595
runtime applications,
creating, 589, 592-593
Solution Options, 590
Bindkey option, 591
Closing Splash Screen
option, 592
Custom Image
option, 592
Extension option, 591
Runtime Name
option, 591
solutions, customizing, 594

dialog windows

designing, 416, 418-419
pause states, 419

DIF (Data Interchange Format), exporting

data, 516

**digital cameras, photo
imports, 505-506**

directories

AutoUpdate directory
building, 626
folder naming
structures, 627
FileMaker Server mainte-
nance, 634
phone directory flat file,
example of, 12

**displaying records in layouts,
troubleshooting, 141**

**Distribute command
(Arrange menu), 132**

Div functions, 218

**Do Not Evaluate If All
Referenced Fields Are
Empty check box (Specify
Calculation dialog),
204-205**

**Do Not Store Calculation
Results check box
(Indexing option), 206**

**documentation, DDR,
446-447**

documenting solutions, 457

**downloading plug-ins,
631-632**

**driver managers, ODBC,
472-473**

drivers, ODBC, 472-473
installing, 474-476
managing, 478

**Drop-Down Calendar option
(Field/Control Setup
dialog), 136**

**Drop-down list formats
(fields), 46**

**Drop-Down List option
(Field/Control Setup
dialog), 136**

DSN, configuring
for Mac OS X, 479, 482
for Windows OS, 483-484

-dup command, 569

**Duplicate command (Edit
menu), 125**

duplicating

found record sets during
imports, 510
layouts, 107

durability (ACID tests), 289

dynamic reports, 270-273

E

**Edit Account dialog (Manage
Accounts & Privileges dia-
log), 305**

**Edit Box formats (fields),
46, 136**

-edit command, 569

**Edit Custom Function
dialog, 374-375**

Edit menu (Layout mode)
Duplicate command, 125
Select All command, 126

Edit Menu Set dialog, 344

Edit mode (IWP), 543

**Edit Privilege Set dialog
(Manage Accounts &
Privileges dialog), 307**

**Edit Privileges dialog
(Manage Accounts &
Privileges dialog)**
Allow Exporting option, 313
Allow Printing option, 313
Allow User to Modify Their
Own Password option, 313
Available Menu Commands
option, 313
Custom Value List
Privileges option, 311
Data Access and Design
option, 312
Layouts drop-down list, 310
All Modifiable
setting, 310
All No Access
setting, 310

- All View Only
 - setting, 310
- Custom Privileges
 - setting, 310
- Manage Extended Privileges
 - option, 315-316
- Edit Relationship dialog (Relationships Graph), 168, 178**
- Edit Script dialog, 233**
- Edit setting (Custom Privileges dialog), 309**
- editing**
 - custom menus (user interfaces), 345
 - menu sets, assigning, 346
 - menu sets, loading/activating, 346-347
 - data, audit trails, 295
 - fields, 45
 - scripts, 233
 - value lists (fields), 47-48
- Else If script steps, 247-248**
- Else script steps, 247-248**
- email, sending as reports, 281**
- embedding query parameters in stylesheets, 566-567**
- employee tables, example of, 12**
- Enable Account script step, 323**
- enabling**
 - AutoUpdate plug-in, 628
 - Kiosk mode via Developer Utilities (FileMaker Developer), 595-596
 - plug-ins, 599
- encoding URL parameters, 570**
- End If script steps, 247-248**
- ending IWP sessions**
 - FileMaker Pro 8
 - configuration, 526
 - troubleshooting, 544
- entities, 144-146**
 - candidate entities, 147
 - child entities, 150
 - join entities, 154
 - join tables, 154
 - keys, 151
 - foreign keys, 152
 - primary keys, 152, 164
 - parent entities, 150
 - relationships, 148
 - many-to-many relationships, 151, 154, 156-157
 - many-to-one relationships, 150
 - notation, 148-149
 - one-to-many relationships, 149-152, 160-161
 - one-to-one relationships, 150
- Envelope layouts, 106**
- equijoins, 20, 178**
- ER diagrams, 457**
- ERD (entity-relationship diagrams), 144**
 - attributes/entities, 147
 - many-to-many relationships, 154
 - notation, 148-149
 - one-to-one relationships, 149
 - Relationships Graph, 177
- error capturing**
 - troubleshooting, 295
 - unsupported script steps (IWP), 534
- error codes, FMSAUC_UpdatePlugin (AutoUpdate plug-in), 629. See also error messages**
- error handling, scripts, 432**
 - Get(LastError)
 - function, 433
 - reproducing errors, 434
 - Set Error Capture
 - scripts, 433
- error logs, Developer Utilities (FileMaker Developer), 594-595**
- error management scripts, 239-240**
 - Allow User Abort scripts, 240
 - Get (LastError)
 - functions, 240
 - Set Error Capture
 - scripts, 240-241
- error messages**
 - locked records, 292
 - no records found, 554
 - scripts, troubleshooting, 253
- escaping, 391**
- Evaluate functions, 356**
 - multi-valued script parameters, passing, 391
 - subsummary reports, 270
 - uses of, 357-358
- EvaluationError functions, 358**
- event handlers, 405**
- event logs, FileMaker Server maintenance, 635**
- event triggers, 404**
 - for layouts, 406
 - for objects, 407
 - targets, 404
- Exact functions, 213**
- exact match searches (CWP URL searches), 554-555**
- Excel documents**
 - exporting data, 515
 - sending as reports, 280
 - saving in FileMaker Pro, 69
- executing scripts, 233, 251**
- exit conditions (loops), 249-250**
- Exit script, 236**
- exiting**
 - IWP sessions, 526
 - loops, 250
- Export Field Contents option (container fields), 51**

exporting data

- character transformations, 514
- container fields, 518
- fields, selecting, 512-514
- file formats
 - choosing, 512
 - CSV format, 515
 - DBF file format, 516
 - DIF, 516
 - Excel file format, 515
 - FileMaker Pro file format, 515
 - HTML Table format, 515
 - Merge format, 515
 - tab-separated text format, 515
 - WKS file format, 516
 - XML file format, 515
- file privileges, 513
- fixed-width formats, padding data, 517-518
- formatting, 516
- from all records, 514
- grouped data, 517
- large fields, 518
- privileges, 313
- reasons for, 511
- related fields, 516
- scripts, 519
- source tables, choosing, 512

Extend Found Sets, 63**Extend function, 204****extend requests, 63****extended privileges, 529**

- Access via FileMaker
 - Mobile privilege, 315
- Access via FileMaker
 - Network privilege, 314
- Access via Instant Web
 - Publishing privilege, 314
- Access via ODBC/JDBC
 - privilege, 314
- Access via XML Web
 - Publishing privilege, 315
- Access via XSLT Web
 - Publishing privilege, 315
 - configuring for PHP, 576-577

- CWP, database configuration, 548-549
- user-level internal security
 - custom privileges, 315-316
 - default privileges, 314-315

Extended Privileges tab (Accounts and Privileges menu), 529**Extension option (Solution Options), 591****external authentication, 319-320****external authentication feature (FileMaker Server), 604****external data sources**

- fixing, 458
- flat-file sources, 493
 - Import Field Mapping dialog, 494-497
 - target, selecting, 494
- in multifile table
 - systems, 188
- Microsoft Excel file, importing data, 500
- ODBC
 - adding to Relationships Graph, 486
 - specifying, 485-488

F**fact tables, 157****Feiler, Jesse, 508****Field Access setting (Custom Privileges dialog), 309****Field Behavior dialog**

- accessing, 137
- field access, configuring, 137
- Go to Next Object Using option, 137

field definition dialog, Comment dialog, 430**Field List feature (Specify Calculation dialog), 85, 201****field list filtering, 513****field steps (scripts), 241**

- conditional script steps, 247-250
- Copy, 242
- Cut, 242
- Else If script steps, 247-248
- Else script steps, 247-248
- End If script steps, 247-248
- Find, 245
- Go to Field, 242
- Go to Layout, user navigation scripts, 243
- If script steps, 247-248
- My Set Field, troubleshooting, 254
- Open Record/Request, 292
- Paste, 242
- Set Field, 242-243
- Show Custom Dialog script steps, 250
- Sort, 246-247

-field URL parameters, 570**field validation errors, troubleshooting, 509****Field/Control Setup dialog, 136****fieldname URL parameters, 570****Fieldname.op URL parameters, 570****fields, 31**

- access, configuring, 137
- appearance of, 45
- as buttons, 136
- auto-entry options, 89
 - Calculated Value option, 92
- calculation formulas, 200
- Creation option, 90
- Data option, 91
- Looked-Up Value option, 92-93
- Modification option, 90
- Serial Number option, 90-91

- Value from Last Visited
 - Record option, 91
 - behaviors, defining, 137
 - calculated replaces, calculated formulas, 200
 - calculation fields, 49, 85-86, 197
 - data types, 204
 - data types, troubleshooting, 226
 - global storage, 205
 - indexing, 206-207
 - number of repetitions, 204
 - comments
 - adding, 81
 - proactive troubleshooting, 430
 - container fields, 49, 84-85
 - Export Field Contents option, 51
 - exporting data, 518
 - Import option, 50
 - Insert option, 50
 - OLE support, 50
 - Paste option, 50
 - saving/retrieving information in, 50
 - copying/pasting, troubleshooting, 141
 - creating, 82
 - Creation Account Names fields, 93
 - Creation Timestamp fields, 93
 - data entry, 72-73, 137
 - date fields, 49, 83
 - developer fields, 80, 429
 - editing, 45
 - exporting data, selecting for, 512-514
 - formatting
 - Check box set formats, 46
 - data in, 47
 - Drop-down list formats, 46
 - Edit box formats, 46
 - Pop-up calendar formats, 46
 - Pop-up menu formats, 46
 - Radio button set formats, 46
 - Furigana, 100
 - globals, 51, 96
 - login accounts, 287
 - troubleshooting, 294
 - indexing, 97
 - unindexable fields, 100
 - value indexes, 98-99
 - word indexes, 98
 - large fields, exporting data, 518
 - layouts, 135-137
 - matches, building, 178
 - merge fields, adding to layouts, 139
 - misspellings, 45
 - Modification Account Names fields, 93
 - Modification Timestamp fields, 93
 - naming conventions, 79-81, 429
 - navigating, 46, 138
 - number fields, 49, 83
 - numerical fields, numerical comparison searches (CWP URL searches), 555
 - post file conversion tasks, 465-468
 - primary key fields, configuring, 164
 - related fields, exporting, 516
 - repeating fields, 97, 514
 - selecting in calculation formulas, 201
 - serial key fields, creating, 91
 - summary fields, 49, 87-89, 268-270
 - tab order, 138
 - text fields, 49, 83, 93
 - time fields, 49, 83
 - timestamp fields, 49, 84, 93
 - troubleshooting, 100-101
 - unindexed fields, searches/sorts, 436
 - user fields, naming, 429
 - validating, 94-96
 - calculation formulas, 200
 - condition failures, 95
 - troubleshooting, 101
 - value lists
 - editing, 47-48
 - other values, 48
 - sorting, 67
- Fields tab (Manage Database dialog), 76**
- file formats, FileMaker Pro 9 changes, 19**
- File Options dialog, creating default accounts, 304**
- file paths, script variables, 400**
- file references. *See also* files**
- building, 188-190
 - errors, 600
 - fixing, 458
 - search paths, adding, 189
- file requests on multiple layouts, 64**
- file-level access security, 316**
- external authentication, 319-320
 - file list filtering, 321
 - networks, 317
 - server administration, 316
 - user authentication, 318
- FileMaker Developer Developer Utilities**
- creating runtime applications, 589, 592-593
 - customizing solutions, 594
 - enabling Kiosk mode, 595-596
 - error logs, 594-595
 - removing Admin access, 593
 - renaming files, 588
 - securing Kiosk mode, 595
 - Solution Options, 590-591
 - sample plug-in, functions of, 598

FileMaker Mobile, Access via FileMaker Mobile extended privilege, 315

FileMaker Network, Access via FileMaker Network extended privilege, 314

FileMaker Pro 6, FileMaker Pro 9 feature comparison, 22

FileMaker Pro 8, IWP configuration, 524-526

FileMaker Pro 10
 architectural changes
 file formats, 19
 files with multiple windows, 21
 privilege sets, 21
 relationship functionality, 20
 Relationships Graph, 19
 FileMaker Pro 6 feature comparison, 22
 functionality, extending, 17
 new features, overview of, 18
 overview of, 14
 security, 21

FileMaker Pro 10 Advanced, 15-17

FileMaker Pro file format (exporting data), 515

FileMaker Server, 601
 access, troubleshooting, 573
 Auto Update feature, 626
 configuration, selecting, 610
 Consistency Checker, 603
 database hosting, 616
 deploying, 611-612, 616
 disconnecting from, 313
 external authentication feature, 604
 installing, 608-609
 maintenance tips
 database directory structures, 634
 hardware requirements, 633
 monitoring event logs, 635

monitoring usage statistics, 634
 network infrastructures, 633
 scheduling backups, 634
 server configuration, 634
 software installation, 633
 software updates, 635
 performance, troubleshooting, 436-437
 plug-in management feature, 604
 plug-ins
 automatically updating, 626, 627-632
 deploying, 599
 requirements for, 604-606
 SAT, 603
 SSL data transfer feature, 604
 starting/stopping, 616
 versus peer-to-peer database hosting, 602-603

FileMaker Server 8 Advanced, 527

FileMaker Server 9, 15

FileMaker Server 9 Advanced, 15

FileMaker Server Published Databases page (WPAC), 527

FileMaker-to-FileMaker imports, 509-510

files

lists, filtering, 321
 multiple windows, FileMaker Pro 9 changes, 21
 open files, backups, 603
 recovering, 443-446
 renaming
 Developer Utilities (FileMaker Developer), 588
 file reference errors, 600
 runtime applications, adding/replacing, 591
 sharing, IWP, 530
 viewing in Web Viewer, 332

Filter functions, 368-369

filtering
 field lists, 513
 file lists, 321

filtering functions
 Filter functions, 368-369
 FilterValues functions, 369-370

FilterValues functions, 369-370

-find command, 554-555, 569

find layouts, 423-424

Find mode, 32
 dedicated layouts, 423
 find requests, 58
 constrain requests, 63
 extend requests, 63
 multiple find requests, 62
 on multiple layouts, 64
 search symbols, 62
 wildcard searches, 62
 Modify Last Find feature, 63

Find mode (IWP), 543

find requests, 58
 Modify Last Find feature (FileMaker Pro Find mode), 63
 multiple find requests, 62
 search symbols, 62
 wildcard searches, 62

Find script step, 245

-findall command, 569

-findany command, 569

finding
 records, CWP URL searches, 554
 tables, CWP URL searches, 553

firewalls

CWP URL requests, 573
 ports for FileMaker Server installation, 609
 server connections, troubleshooting, 439

fixed-width formats, exporting to, 517-518

fixing file references, 458

- flat-file data sources,**
 - 10, 14, 493
 - importing, 494-497
 - phone directory flat files, example of, 12
- Floor functions, 216-217**
- FMSAUC_FindPlugin function (AutoUpdate plug-in), 628-630**
- FMSAUC_UpdatePlugin function (AutoUpdate plug-in)**
 - error codes, 629
 - plug-ins, downloading, 631-632
- FMSAUC_Version function (AutoUpdate plug-in), 628**
- folder imports, 501**
 - image file imports, 503-505
 - text file imports, 502-503
- folders, naming AutoUpdate directory folders, 627**
- footers, 117, 537**
- foreign keys (entities), 152**
- Form view (FileMaker Pro), 33**
- Format Painter tool, 124**
- formatting**
 - conditional formatting, 348
 - exported data, 516
 - fields, 46-47
 - files, FileMaker Pro 9
 - changes, 19
 - layout fields, 136-137
 - layout parts, 118
 - reports, 262
- formatting text functions, 362**
 - RGB, 363
 - TextColor, 363
 - TextColorRemove, 364
 - TextFont, 363
 - TextFontRemove, 364
 - TextFormatRemove, 364
 - TextSize, 363
 - TextSizeRemove, 364
 - TextStyleAdd, 363
 - TextStyleRemove, 363
- forms, viewing as layouts, 111**
- Formula box (Specify Calculation dialog), 86, 200**
- formulas**
 - Boolean formulas, 199
 - calculation formulas, 197
 - auto-entry options (fields), 200
 - calculated replaces (fields), 200
 - debugging, 353
 - field names, troubleshooting, 225-226
 - field validation, 200
 - record-level security, 200
 - script steps, 199
 - specifying context, 207-209
 - writing, 198, 200-202
 - complex formulas, simplifying, 371-372
 - post file conversion tasks, 465-468
- found sets, 56, 58**
 - printing, 67
 - window construction, troubleshooting, 424
- full access privileges (scripts), 234**
- Full Access privilege set option (Manage Accounts & Privileges dialog), 307**
- Function List (Specify Calculation dialog), 202**
- Function list feature (Specify Calculation dialog), 86**
- function palettes (multi-window interfaces), 415**
- functions**
 - Abs, 218
 - aggregate functions, 221-222
 - array functions, 364
 - GetValue functions, 366
 - LeftValue functions, 365, 383
 - MiddleValues functions, 365, 383
 - RightValues functions, 366, 383
 - stepping through, 366
 - usage examples, 366
 - ValueCount functions, 365, 383
- calculation functions**
 - Exact functions, 213
 - Length functions, 212
 - PatternCount functions, 212
 - Position functions, 213
 - Substitute functions, 214
 - Trim functions, 213-214
 - WordCount functions, 213
- case altering, 214**
- Ceiling, 217**
- components of, 210**
- conditional functions, 220-221**
- Count functions, 222**
- custom functions, 370**
 - automatic updates, 373
 - building, 373-375
 - comments, 430
 - configuring
 - availability, 375
 - CrossProduct functions, 379-380
 - custom function libraries, building, 384-385
 - defining system constants, 372
 - Hypotenuse functions, 376
 - naming conventions, 374, 430
 - naming parameters, 430
 - NthRoot functions, 376
 - PadCharacters functions, 378
 - Quarter functions, 376
 - recursive functions, 373
 - refactoring, 372
 - RepeatText functions, 377
 - restricting access, 375
 - simplifying complex formulas, 371-372

- TrimChar functions, 379
- WeekEndingFriday functions, 376
- DatabaseNames functions, 225
- Date functions, 218-220
- Design functions, 224-225
- Div, 218
- Evaluate
 - passing multi-valued script parameters, 391
 - subsummary reports, 270
- EvaluationError functions, 358
- Exact, 213
- Extend, 204
- filtering functions
 - Filter functions, 368-369
 - FilterValues functions, 369-370
- Floor function, 216-217
- Get functions, 222-224
- Get(LastError), 433
- Get(ScriptParameter), retrieving script parameter values, 389
- Get(ScriptResult), 395
- GetAsBoolean, 199
- GetLayoutObjectAttribute, identifying current Web Viewer page, 331-332
- GetNthRecord functions, 381-382
- GetParam, passing multi-valued script parameters, 392
- GetSummary, subsummary reports, 269-270
- Int, 216-217
- IsValidExpression functions, 358
- Length, 211-212
- Let, passing multi-valued script parameters, 390-391
- logical functions
 - Choose functions, 354-355
 - Evaluate functions, 356-358
 - GetField functions, 356
 - Let functions, 351-353, 383

- Lookup functions, 359
- LookupNext functions, 359-361
- Self function, 409-410
- Mod, 218
- naming, 430
- nested functions, 215-216
- nesting functions, 211
- number functions, 216
 - Abs functions, 218
 - Ceiling functions, 217
 - Div functions, 218
 - Floor functions, 216-217
 - Int functions, 216-217
 - Mod functions, 218
 - Random functions, 218
 - Round functions, 217
 - Truncate functions, 217
- parameters, 210, 430
- PatternCount, 212
- plug-in functions, 597
- Position, 213
- private, naming conventions, 375
- public, 375
- Random, 218
- Round, 217
- selecting in calculation formulas, 202
- Substitute, 214
- text formatting functions, 362
 - nontext calculations, troubleshooting in, 382
- RGB functions, 363
- TextColor functions, 363
- TextColorRemove functions, 364
- TextFont functions, 363
- TextFontRemove functions, 364
- TextFormatRemove functions, 364
- TextSize functions, 363
- TextSizeRemove functions, 364
- TextStyleAdd functions, 363
- TextStyleRemove functions, 363

- text functions, 212
 - case altering functions, 214
 - Exact functions, 213
 - Length functions, 212
 - PatternCount functions, 212
 - Position functions, 213
 - Substitute functions, 214
 - text parsing functions, 215
 - Trim functions, 213-214
 - WordCount functions, 213
- text parsing, 215
- Time functions, 218-220
- TimeStamp, 219-220
- Trim, 213-214
- Truncate, 217
- ValueListItems functions, 225
- WindowNames functions, 225, 413
- WordCount, 213
- XMpl-Add, 598
- XMpl-Append, 598
- XMpl-NumToWords, 598
- XMpl-StartScript, 598
- XMpl-UserFormat Number, 598

Furigana, 100

G

general slowness (performance), troubleshooting, 436

generic report structures, versus specific report structures, 259

Get functions, 222-224

- Get(ApplicationVersion) function, IWP execution tests, 535
- Get(LastError) function, 240, 433
- Get(LastMessageChoice) function, 250
- Get(ScriptParameter) function, retrieving script parameter values, 389
- Get(ScriptResult) function, 395

GetAsBoolean function, 199
GetField functions, 356
GetLayoutObjectAttribute function, identifying current Web Viewer page, 331-332
GetNthRecord functions, 381-382
GetParam functions, passing multi-valued script parameters, 392
GetSummary function, sub-summary reports, 269-270
GetValue functions, 366
 global fields, 51
 Global Storage option (Storage Options dialog), 205
 global variables, 399-400
 globally replacing data, 71-72
 globally stored fields
 global variables versus, 399
 uses for, 400
 globals, 96, 287
 calculation fields, 205
 default values, troubleshooting, 294
 login accounts, 287
 troubleshooting, 442-443
Go to Field script steps, 242
Go to Layout script step, user navigation scripts, 243
Go to Next Object Using option (Field Behavior dialog), 137
Go to Related Record Options dialog, 421
Go to Related Record scripts, 420-422
Grammar section (URL), 552
 -grammar URL parameters, 570
Graphic menu command (Insert menu), 123

graphics
 IWP layout design restrictions, 536
 layouts, adding to, 123
 grouped data, exporting, 517
 grouping/ungrouping layout objects, 130
GTRR (Go To Related Records) scripts, 420-422, 438
 Guest accounts, user-level internal security, 304

H

hard drives, FileMaker Server requirements, 605
 hardware requirements for FileMaker Server, 633
 headers/footers, 116
 IWP layout design restrictions, 537
 sortable headers, 276
Help menu, Resource Center command, 29
hiding
 layouts, 114
 report elements prior to printing, 262
 scripts, 238
 Status toolbar (IWP), application flow, 540
highlighting portal rows, troubleshooting, 425
home pages, building IWP home pages, 541
house icon (IWP Status toolbar), 527
HTML (HyperText Markup Language)
 DDR, 447
 search result stylesheets, building, 565-566
HTML Table format (exporting data), 515
hyphens (-), layout names, 115
Hypotenuse functions, 376

I

identifying
 current page in Web Viewer, 331-332
 unsupported script steps, 533
identity functions (plug-ins), 630-631
If - Else If script step, 292
If script steps, 247-248
image files
 importing, 503-504
 references, importing, 504
 thumbnails, importing, 504-505
imperative programming, 471
Import option (container fields), 50
importing
 data imports, troubleshooting, 508-509
 existing records, updating with imported data, 497-498
 field validation, troubleshooting, 509
 found record sets, duplicating, 510
 from FileMaker Pro file, 498
 from flat-file sources, 494
 Import Field Mapping dialog, 494-497
 target, selecting, 494
 from Microsoft Excel file, 500
 image files, 503-505
 layouts, 107
 ODBC data into FileMaker, 484-485
 photos from digital cameras, 505-506

- scripted imports, 506-507
- scripts, 233, 238
- table records, table
 - context, 440
 - text files, 502-503
- incomplete scripts, troubleshooting, 253**
- indexing**
 - calculation fields, 206-207
 - fields, 97
 - unindexable fields, 100
 - value indexes, 98-99
 - word indexes, 98
- Indexing option (Storage Options dialog), 206-207**
- Indicate Web Compatibility check box (ScriptMaker), 533**
- InitializeGlobals scripts, 255**
- initiating scripts, 251**
- Insert menu (Layout mode), 72, 123**
 - Merge Field command, 139
 - Graphic menu
 - command, 123
- Insert option (container fields), 50**
- Insert, From Index command (FileMaker Pro), 72**
- inserting data into fields, 72-73**
- installing**
 - FileMaker Server, 608-609
 - ODBC drivers, 474-476
 - plug-ins, 598
 - software, FileMaker Server
 - maintenance tips, 633
- Instant Web Publishing setup screen**
 - accessing, 524
 - database files, extended
 - privileges, 529
- Int functions, 216-217**
- integrity (data), troubleshooting, 70**
- interfaces, creating for**
 - many-to-many relationships, 174-175**
 - interfaces (user)**
 - building, 336-337
 - custom menus
 - components of, 341
 - editing, 345
 - FileMaker control
 - of, 342
 - managing, 342-344
 - menu sets interface, 342-344
 - menu sets, assigning, 346
 - menu sets, loading/
 - activating, 346-347
 - scratch-building, 348
 - uses of, 341
 - dedicated find layouts, 423-424
 - design guidelines, 349-350
 - look/feel, designing, 338-339
 - multi-window interfaces
 - dialog windows, 416-419
 - tool/function
 - palettes, 415
 - uses of, 415
 - native user interface, 336
 - optimizing, 337
 - prototype layouts/
 - menus, 335
 - security plans, 301
 - UI files, 339-340
 - internal security (user-level)**
 - extended privileges
 - custom privileges, 315-316
 - default privileges, 314-315
 - privilege sets, 306-307
 - conditional
 - privileges, 310
 - data access, controlling, 308-309
 - data validation warning
 - override privileges, 313
 - export privileges, 313
 - FileMaker Server
 - disconnects, 313
 - layout use/
 - development, 310-311
 - menu commands
 - access, 313
 - passwords, setting, 313
 - printing privileges, 313
 - run script privileges, 312
 - value list access, 311-312
 - user accounts, 303-306
- interrogating text strings, 212**
- IP addresses, wildcards, 525**
- isolation (ACID tests), 289**
- IsValidExpression functions, 358**
- iterative process, database design as, 148**
- IWP (Instant Web Publishing), 545**
 - Allow User Abort
 - setting, 534
 - application flow, 538
 - creating file links, 541
 - explicit record
 - commits, 539
 - hiding Status toolbar, 540
 - portals, 540
 - session management, 538
 - Browse mode, 542
 - Commit Record/
 - Request step, 535
 - constraints, 532-533
 - container field restrictions, 537-538
 - data, importing/
 - exporting, 532
 - data-entry, 532
 - database development
 - tools, 532
 - Database Homepage, 530
 - defining, 521
 - deploying, 523
 - Edit mode, 543
 - ending sessions,
 - troubleshooting, 544
 - file security, 532
 - authenticating
 - accounts/privileges, 531
 - extended privileges, 529
 - file sharing, 530

FileMaker Pro 8
 configuration, 524-526
 FileMaker Pro keyboard
 shortcuts, 532
 FileMaker Pro toolbars, 533
 FileMaker Server 8
 Advanced
 configuration, 527
 Find mode, 543
 goal of, 521
 graphical layout
 elements, 533
 home pages, building, 541
 IWP controls, Log Out
 button, 526
 layout design restrictions,
 535-537
 passwords, 533
 Preview mode, 532
 script support, 533-535
 Show/Hide Status toolbar
 script step, 539
 Sort dialog, 544
 spell-checking, 533
 Status toolbar
 Commit button, 539
 house icon, 527
 triggers, 406
 turning on/off, 524
 unsupported script steps
 error capturing, 534
 identifying, 533
 value lists, 533
 versus CWP, 546-547
 window manipulation
 tools/techniques, 533

**IWP Database Homepage,
 Advanced Web Publishing
 Options dialog, 524**

J

JavaScript, disabling, 523
 join entities, attributes, 154
 join tables, 154, 428
 joins
 equijoins, 20
 join tables, naming, 428

K

key fields, 13, 429
keyboard shortcuts
 FileMaker Pro, 71
 IWP, 532
keys (entities), 151
 foreign keys, 152
 match fields, 164
 primary keys, 152, 164
**keystrokes, event handler
 chain, 405**
**Kiosk mode (FileMaker Pro
 Advanced), 17**
 enabling via Developer
 Utilities (FileMaker
 Developer), 595-596
 securing via Developer
 Utilities (FileMaker
 Developer), 595
kiosks, 589

L

Label layouts, 106
**LANs, opening databases
 on, 42**
**language support,
 Furigana, 100**
**large fields, exporting
 data, 518**
launch files, 294
launching
 Admin Console, 617
 PHP Site Assistant, 580
 XSLT Site Assistant, 559
-lay URL parameters, 570
**-lay.response URL
 parameters, 570**
layering layout objects, 131
**Layout bar (Status
 toolbar), 120**
layout group, selecting, 581
Layout mode, 33
 accessing, 104

Always Lock Layout
 tools option, 123
 Arrange menu
 Align command, 132
 Bring Forward
 command, 131
 Bring to Front
 command, 131
 Distribute
 command, 132
 Object Grids
 command, 127
 Resize To alignment
 tools, 127
 Resize To command, 132
 Send Backward
 command, 131
 Send to Back
 command, 131
 columns, boundaries, 113
 dependencies, table
 context, 441
 Edit menu
 Duplicate command, 125
 Select All command, 126
 Insert menu, 123
 New Layout/Report
 wizard, 262-265
**layout objects. See also
 layouts**
 aligning, 132
 attributes, specifying, 123
 centering, 132
 default format attributes,
 setting, 124
 duplicating, 125
 Format Painter tool, 124
 grouping/ungrouping, 130
 layering, 131
 locking/unlocking, 131
 moving, 126
 naming, 129
 object grids, 127-128
 positioning, 125-126
 resizing, 126, 129-130
 selecting, 125-126
 setting default format attrib-
 utes, 124
 Size palette, 128-129
 sliding, 277-278
 stacking orders, 131

Layout Setup dialog

accessing, 110
 Printing tab, 113, 128
 Views tab, 113

Layout tools (Status toolbar), 122**-layoutnames command, 569****layouts, 31, 103**

Blank Layouts, 106
 bodies, 116, 119
 buttons, building, 252-253
 Columnar List/Report layouts, 106
 context, 108, 439
 creating, 105-107
 cross-platform layouts, building, 142
 dedicated find layouts, 423-424
 duplicating, 107
 Envelope layouts, 106
 fields
 adding, 135-136
 formatting, 136-137
 navigating, 138
 FileMaker Pro layouts, 32
 Footers, 117
 Format Painter tool, 124
 Headers, 116
 headers/footers, IWP restrictions, 537
 hiding, 114
 importing, 107
 IWP restrictions, 535
 graphical elements, 536
 headers/footers, 537
 layout part size, 537
 View As options, 536
 Label layouts, 106
 Leading Grand Summaries, 116
 list view layouts, 262
 merge fields, adding to, 139
 multi-column layouts, 113-114
 multiple layouts
 find requests, 64
 working with FileMaker Pro, 73
 naming conventions, 115-116, 429

part size, IWP restrictions, 537

parts

adding, 117
 defining, 118-119
 deleting, 118
 formatting, 118
 setting background colors, 118
 sizing, 118

PHP, reviewing, 577

pictures, adding, 123

post file conversion

tasks, 468

printing, sliding layout

objects, 277-278

records, displaying, 141

reordering, 115, 117

Standard Form layouts, 106

subsummaries, 116, 119

Tab Control objects,

adding to, 134

table context, 440, 553

table occurrences, 108-109

Table View layouts, 106

Title Footers, 117

Title Headers, 116

tooltips, adding, 140

Trailing Grand

Summaries, 116

triggers, 406

usage/development privileges, security, 310-311

user interfaces, prototype layouts, 335

view options, 111

forms view, 111

lists view, 112

restricting user

access, 113

tables view, 112

Layouts drop-down list (Edit Privileges dialog), 310

All Modifiable setting, 310

All No Access setting, 310

All View Only setting, 310

Custom Privileges

setting, 310

Leading Grand

Summaries, 116

Left to Right alignment option (Align command), 132**LeftValue functions, 365, 383****Length function, 211-212****Let functions, 351-352**

calculation formulas,

debugging, 353

multi-valued script parameters, passing, 390-391

variable names,

troubleshooting, 383

variables, using multiple instances of, 353

libraries, building script libraries, 255**linking Web pages to IWP files, 541****list reports, designing**

alternating row color, 274

sortable column

headers, 276

List view (FileMaker Pro), 33**list view layouts, 262****lists, viewing as layouts, 112****live reporting, 270-273****local files, opening in FileMaker Pro, 40****local variables, 397**

scope, 398

uses for, 400

locking

layout objects, 131

multiple contexts

(scripts), 293

multiple windows, 292-293

records, 291-292

Log In Using option (File Options dialog), creating default accounts, 304**Log Out button (IWP controls), 526****logical functions**

Choose functions, 354-355

Evaluate functions, 356-358

GetField functions, 356

- Let functions, 351-353
 - debugging calculation formulas, 353
 - troubleshooting, 383
 - variables, using multiple instances of, 353
- Lookup functions, 359
- LookupNext functions, 359-361
- Self function, 409-410
- logical operators, 202, 556**
- logins**
 - automatic logins, user-level security, 304
 - globals, 287
 - testing, 294
 - troubleshooting, 294
- Looked-Up Value option (field auto-entry options), 92-93**
- Lookup functions, 359**
- LookupNext functions, 359-361**
- lookups, 403**
- loop scripts, calculated replaces, 437**
- loops, 248**
 - conditional script steps, 250
 - exit conditions, 249-250
 - exiting, 250
 - testing, 254
- lop command, 556**
- lop URL parameters, 570**
- lost data, troubleshooting, 70**

M

- Mac OS X**
 - ODBC administration, 477
 - DSN, configuring, 479, 482
 - data source names, 477
 - drivers, 478
 - plug-ins, Windows servers, 627
 - Status toolbar customizing, 37-38

- maintaining**
 - FileMaker Server
 - database directory structures, 634
 - hardware
 - requirements, 633
 - monitoring event logs, 635
 - monitoring usage statistics, 634
 - network
 - infrastructures, 633
 - scheduling backups, 634
 - server configuration, 634
 - software installation, 633
 - software updates, 635
 - sort order, 270

- Manage Accounts & Privileges dialog**
 - Accounts tab
 - user account privilege sets, reviewing, 305
 - user account security, 303
 - Edit Account dialog, security, 305
 - Full Access privilege set option, 307
 - Privilege Sets tab, 306-309

- Manage Custom Menus dialog, 342**

- Manage Database dialog, 75-76**

- Comment text box, 81
- Fields tab, 76
- Relationships Graph, Specify Table dialog, 180
- Relationships tab, 76
- Table menu, Table Name box, 161

- Manage Extended Privileges option (Edit Privileges dialog), 315-316**

- Manage Scripts dialog (ScriptMaker Interface), 232-233**

- managing**
 - branching scripts, conditional script steps, 247-248
 - custom menus (user interfaces), 342-344

- errors
 - error management scripts, 239-241
 - Get (LastError) function, 240
- Relationships Graph, 193-194
- scripts, 232-233
- sessions, IWP, 538
- windows, 413
- many-to-many relationships, 151, 154, 156-157**
 - interface, designing, 174-175
 - structure, building, 172
 - value lists, creating, 173-174

- many-to-one relationships (entities), 150**

- master records, adding to related records, 169**

- match fields, 164**

- mathematical operators, 202**

- matrixes (security), 299-301**

- max URL parameters, 570**

- memory, FileMaker Server RAM requirements, 605**

- menu items (custom menu components), 341**

- menu sets, interfaces, 342**

- Menu Sets tab (Manage Custom Menus dialog), 342**

- menus**

- commands access, security, 313
- square-bracketed menus, 344
- user interfaces
 - custom menus, 341-348
 - prototype menus, 335

- Merge Field command (Insert menu), 139**

- merge fields, adding to layouts, 139**

- Merge format (exporting data), 515**

- metadata, 9**

Microsoft Excel files,
importing data, 500

MiddleValues functions,
365, 383

migrating. *See* upgrading
FileMaker

misspellings in fields, 45

Mod functions, 218

modal dialog windows
designing, 416-419
pause states, 419

modes, 31
dependencies, table
context, 441

-modid URL
parameters, 570

Modification Account Names
fields, 93

Modification option (field
auto-entry options), 90

Modification Timestamp
fields, 93

Modify Last Find feature
(FileMaker Pro Find
mode), 63

modifying Table views, 271

modular code, 431

modularizing scripts, 393

Move to Back option
(Arrange menu), Tab
Control objects, 134

Move/Resize Window
script, 414

moving
layout objects, 126
scripts, 232

multi-column layouts,
113-114

multi-context locking, 293

multi-user applications
concurrency
ACID tests, 288-289
committed data versus
created data, 290

multi-context locking,
293

multi-window locking,
292-293

record locking, 291-292

script logs, 290

serial IDs, 290

sessions, 286
database tracking, 288
globals, 287, 294
session-specific
elements, 286

multi-user databases
audit trails, auto-entry tech-
niques, 295
launch files, 294

multi-valued script paramet-
ers, passing
Evaluate functions, 391
GetParam functions, 392
Let functions, 390-391
parsing text arrays, 390
structured data
elements, 392

multi-window interfaces
dialog windows, 416-419
tool/function palettes, 415
uses of, 415

multi-window locking,
292-293

multifile table systems, 187
external data sources, 188
file references, building,
188-190

multiple criteria searches
(CWP URL searches), 556

multiple file windows,
FileMaker Pro 10
changes, 21

multiple files, crosstalk, 189

multiple find requests,
62, 556

multiple layouts
file requests, 64
FileMaker Pro, working
with, 73

multiple records, sorting, 66

multiple repetitions, calcula-
tion fields, 204

multiple tables
first tables, building, 160
match fields, 164
portals
adding related records to
master records, 169
viewing related child
data, 165-167
relationships, building, 163
tables, adding to, 161

multitiered pause states, 419

multitiered sorting, 73

multiuser databases, team
development, 295-296

multiple-match relationships
building, 180-183
OR conditions, trou-
bleshooting, 193

My Set Field script step,
troubleshooting, 254

N

name-value pairs, 552-553

naming
AutoUpdate directory fold-
ers, 627
custom functions, 374, 430
database files, 428
fields, 79-81
developer fields, 80, 429
troubleshooting, 101
files
Developer Utilities
(FileMaker Developer),
588
file reference errors, 600
functions, 430
join tables, 428
key fields, 429
layouts, 115-116, 129, 429
parameters, 430
private functions, 375
tables, 77-78
user fields, 429

native user interface, 336

navigating

- array elements, 366
- databases, 73
- fields, 46, 138
- FileMaker Pro layouts, 32

navigation scripts, 243-244

**nested functions, 211,
215-216**

networks

- FileMaker Server connection requirements, 606
- launch files, 294
- security, 317
- team development, 295-296

-new command, 569

**New Layout/Report wizard,
262-265**

**"no records found" error
messages, 554**

non-equijoins, 178-179
troubleshooting, 192-193

**nontext calculations, troubleshooting text formatting
functions, 382**

normalizing data, 155

NthRoot functions, 376

number fields, 49, 83

number functions

- Abs functions, 218
- Ceiling functions, 217
- Div functions, 218
- Floor functions, 216-217
- Int functions, 216-217
- Mod functions, 218
- Random functions, 218
- Round functions, 217
- Truncate functions, 217

**Number of Repetitions
option (Specify Calculation
dialog), 204**

**numerical comparison
searches (CWP URL
searches), 555**

O

object grids, 127-128

**Object Grids command
(Arrange menu), 127**

objects

- layout objects
 - aligning, 132
 - centering, 132
 - duplicating, 125
 - Format Painter tool, 124
 - grouping/
 - ungrouping, 130
 - layering, 131
 - locking/unlocking, 131
 - moving, 126
 - object grids, 127-128
 - positioning, 125-126
 - resizing, 126
 - selecting, 125-126
 - setting default format
 - attributes, 124
 - Size palette, 128-129
 - sliding, 277-278
 - specifying attributes, 123
 - stacking orders, 131
- Tab Control objects, adding
 - layouts to, 134
- triggers, 407

ODBC, 471

- Access via ODBC/JDBC
 - extended privilege, 314
- applications, 472-473
- database configuration, 473-474
- drivers, installing, 474-476
- DSN
 - configuring on Mac OS X, 479, 482
 - configuring on Windows OS, 483-484
- external data sources
 - adding to Relationships Graph, 486
 - specifying, 485-486, 488
- importing data into
 - FileMaker, 484-485
- Mac OS X administration
 - data source names, 477
 - drivers, 478

Windows

- administration, 476
- data source names, 477
- drivers, 478

off-the-shelf software, 10

**OLE support, container
fields, 50**

**Omit Record command
(FileMaker Pro), 64**

**one-to-many relationships
(entities), 149-150**

- building
 - adding tables to multitable systems, 161
 - building first tables in multitable systems, 160
- foreign/primary keys, 152

**one-to-one relationships
(entities), 150**

**Only During Data Entry
option (Validation
dialog), 95**

open files, backups, 603

**Open Record/Request script
step, 292**

**Open Remote File dialog
(FileMaker Pro), 42**

opening

- FileMaker Pro databases
 - on LAN, 42
- local files, 40
- remote files, 41
- files, troubleshooting, 322

operators

- CWP operators, 555
- selecting in calculation formulas, 201

**Operators section (Specify
Calculation dialog), 85, 201**

optimizing interfaces, 337

**OR conditions, troubleshooting multiple-
match relationships, 193**

**organizing relationships,
462-463**

**"other" values (value
lists), 48**

P

PadCharacters functions, 378

padding data (fixed-width formats), 517-518

Page Break Before Each Occurrence option (Part Definition dialog), 119

parameters, 210

- db, 570
- encoding, 570
- field, 570
- fieldname, 570
- Fieldname.op, 570
- grammar, 570
- lay, 570
- lay.response, 570
- lop, 570
- max, 570
- modid, 570
- naming, 430
- nesting functions, 211
- recid, 570
- script, 570
- script parameters
 - modularizing scripts, 393
 - passing, 389
 - passing data between files, 394
 - passing multi-valued parameters, 390-392
 - retrieving, 389
 - specifying, 388
 - usage examples, 388, 393-394
- script-presort, 571
- script.prefind, 571
- skip, 571
- sortfield, 571
- sortorder, 571
- stylehref, 571
- styletype, 571
- token, 571

parent entities, 150, 171

Part Definition dialog, 118-119

Part Setup dialog, 117

passwords

Allow User to Modify Their Own Password option (Edit Privileges dialog), 313

Change Password script step, 323

converted passwords, troubleshooting, 321

forgotten passwords, troubleshooting, 321

IWP, 533

Reset Account Password script step, 323

securing, 302-303

security, 313

Paste option (container fields), 50

Paste script steps, 242

PatternCount functions, 212

pause states, multi-tiered pause states, 419

pause states (dialog windows), 419

PDF documents

reports, saving/sending as, 278

saving in FileMaker Pro, 69

peer-to-peer hosting, 16, 603

Perform button (ScriptMaker Interface), 233

performance, troubleshooting, 435

calculation slowness, 437

general slowness, 436

script slowness, 437-438

searching/sorting slowness, 436

phone directory flat files, example of, 12

photos, importing from digital cameras, 505-506

PHP, 576

extended privileges, configuring, 576

layouts, reviewing, 577

placing files on Web server, 578

preparing FileMaker Server for, 577

PHP Site Assistant, 578

FileMaker Server, connecting to, 580

launching, 580

layout group, selecting, 581

PHP files, specifying location for, 586

site profile selecting, 582

specifying options, 584-585

pictures, adding to layouts, 123

plug-in management feature (FileMaker Server), 604

plug-ins, 596

automatically updating via FileMaker Server, 626-632

AutoUpdate plug-in enabling, 628

FMSUAC_FindPlugin function, 628-630

FMSUAC_UpdatePlugin function, 629, 631-632

FMSUAC_Version function, 628

configuring, 599

deploying via FileMaker Server, 599

downloading, 631-632

enabling, 599

identity functions, 630-631

installing, 598

Mac plug-ins for Windows servers, 627

names, accessing, 630-631

plug-in functions, 597

sample FileMaker Developer plug-in, functions of, 598

troubleshooting, 600

uses of, 597

version numbers, accessing, 630-631

writing, 596

plus sign (+) operators, 202

Pop-up calendar formats
(fields), 46

Pop-up menu formats
(fields), 46

Pop-Up Menu option
(Field/Control Setup dia-
log), 136

pop-up windows, trou-
bleshooting, 424

port numbers, IWP configu-
ration, 525

Port section (URL), 552

Portal Setup dialog, 166-167

portals

- configuring, 166
- IWP application flow, 540
- record locking, 291
- related child data, viewing,
165-167
- related records, adding to
master records, 169
- repeating portals, trou-
bleshooting, 176
- rows
 - creating, 55
 - deleting, 56
 - highlighting, trou-
bleshooting, 425
 - sorting, 56

Position functions, 213

positioning layout objects,
125-126

pre-Filemaker 7 versions,
upgrading, 453

Preferences, comparing
OS X and Windows
platforms, 346

preparing FileMaker Server
for PHP, 577

preventing corrupt files, 443

Preview mode, 32

- Copy command, 441
- IWP, 532
- multi-column layouts, 114
- Set Sliding/Printing dialog,
viewing effects in, 278

primary keys (entities),
152, 164

Print button (ScriptMaker
Interface), 233

printing

- found sets, 67
- layouts, sliding layout
objects, 277-278
- privileges, security, 313
- records, 67
- reports
 - formatting, setting, 262
 - troubleshooting, 282

Printing tab (Layout Setup
dialog), 113, 128

PrintSetUp_landscape
scripts, 255

PrintSetUp_portrait
scripts, 255

private functions, naming
conventions, 375

privilege sets, 21. *See also*
privileges

- customizing, 308-309
- full access, 234
- multiple files, 322-323
- reviewing, 305
- user-level internal security,
306-307
 - conditional
 - privileges, 310
 - data access, controlling,
308-309
 - data validation override
privileges, 313
 - export privileges, 313
 - FileMaker Server
 - disconnects, 313
 - layout use/development,
310-311
 - menu commands
 - access, 313
 - passwords, setting, 313
 - printing privileges, 313
 - run script privileges, 312
 - value list access, 311-312

Privilege Sets tab (Accounts
and Privileges menu),
306, 529

- Custom Privileges dialog,
308-309
- Edit Privilege Set
dialog, 307
- Edit Privileges dialog
 - Allow Exporting
dialog, 313
 - Allow Printing
dialog, 313
 - Allow User to Modify
Their Own Password
option, 313
- Available Menu
 - Commands option, 313
- Custom Value List
 - Privileges dialog, 311
- Data Access and Design
dialog, 312
- Layouts drop-down
list, 310
- Manage Extended
Privileges option,
315-316
- Records setting, 310
- Full Access privilege set
option, 307

privileges

- data, exporting, 513
- extended privileges
 - Access via FileMaker
 - Mobile privilege, 315
 - Access via FileMaker
 - Network privilege, 314
 - Access via Instant Web
Publishing
 - privilege, 314
 - Access via ODBC/JDBC
privilege, 314
 - Access via XML Web
Publishing
 - privilege, 315
 - Access via XSLT Web
Publishing
 - privilege, 315
 - user-level internal
security, 314-316
- troubleshooting, 321

proactive troubleshooting
 calculations, debugging, 434
 code, writing
 comments, 430-431
 naming database files, 428
 naming fields, 429
 naming functions, 430
 naming layouts, 429
 naming parameters, 430
 failure, planning for, 431
 scripts, debugging, 448-451
 scripts, error handling, 432-434

-process command, 569

processing instructions

processing instructions (XML), 567-567

processors, FileMaker Server requirements, 605

programming, writing code
 comments, 430-431
 database files, naming 428
 fields, naming, 429
 functions, naming, 430
 layouts, naming, 429
 parameters, naming, 430

Protocol section (URL), 552

public functions, 375

Publishing Engine Configuration page (WPAC), 527

Q

Quarter functions, 376

queries, relationships as. *See also query strings*
 multiple-match relationships, 180-181, 183, 193
 nonequijoins, troubleshooting, 192-193
 table occurrences, 180

query strings
 name-value pairs, 552-553
 processing instructions, 567
 XML publishing, 552-553
 XSLT, 566

Query-string section (URL), 552

Quick Start screen, 27-29

quotes, escaping, 391

R

Radio button set formats (fields), 46

RAM (random access memory), FileMaker Server requirements, 605

Random functions, 218

Re-login script step, 323

rearranging subsummary parts, 267

-recid URL parameters, 570

record pointers, table context, 442

records, 31
 child records, 56
 committed records, 44
 creating in FileMaker Pro, 44
 CWP URL searches, 554
 deleting from FileMaker Pro, 44
 exporting data from all records, 514
 found sets, 56, 58
 layouts, displaying in, 141
 locking, 291-292
 multiple records, sorting, 66
 Omit Record command (FileMaker Pro), 64
 portals
 creating/deleting rows, 55-56
 mechanics of, 55
 sorting, 56
 printing, 67
 security, calculation formulas, 200
 Show All Records command (FileMaker Pro), 64

Records setting (Edit Privileges dialog), 310

Recover command, 443

recovering files, 443-446

recursive functions, 373

recursive scripts, 401-402

refactoring, 372

references, importing, 504

registration, 24

related fields, exporting, 516

related records
 master records, adding to, 169
 troubleshooting, 254

relational databases, 10
 designing
 as iterative process, 148
 attributes, 144-147
 entities, 144-147
 entity keys, 151-152
 entity relationships, 148-152
 ERD, 144, 147
 ERD notation, 148-149
 ERD, one-to-one relationships, 149
 equijoins, 20
 example of, 12-13
 key fields, 13

Relationships Graphs
 ERD, 177
 external tables, adding to, 191
 managing, 193-194
 multiple-match relationships, 182-183

relationships, 148
 as queries
 multiple-match relationships, 180-183, 193
 nonequijoins, troubleshooting, 192-193
 table occurrences, 180
 bidirectional, 181
 cross-product relationships, building, 187
 file references, building, 188-190
 FileMaker Pro 9 changes, 20

- many-to-many relationships, 151, 154-157
- many-to-one relationships, 150
- multiple-match relationships
 - building, 180-181, 183
 - OR conditions, troubleshooting, 193
- multiple tables, building in, 163
- non-equi joins, 178-179
- notation, 148-149
- one-to-many relationships, 149-150
 - building, 160-161
 - foreign/primary keys, 152
- one-to-one relationships, 150
- organizing, 462-463
- with global values, building, 187

Relationships Graph, 77

- Add Table Occurrence icon, 191
- cardinality, 164
- Edit Relationship dialog, 168
- FileMaker Pro 9
 - changes, 19
- shadow fields, 489-490
- table occurrences, 163

Relationships Graph (Manage Database dialog), Specify Table dialog, 180

Relationships tab (Manage Database dialog), 76

relative paths, 458

remote files, opening in FileMaker Pro, 41

removing

- Admin access via Developer Utilities (FileMaker Developer), 593
- records from FileMaker Pro, 44

renaming files

- Developer Utilities (FileMaker Developer), 588

- file reference errors, 600

reordering

- layouts, 115-117
- script steps, 234

repeating fields, 97

- calculation fields, 204
- character transformations (exporting data), 514

repeating portals, troubleshooting, 176

RepeatText functions, 377

replacing

- global data replacements, 71-72
- files in runtime applications, 591

reports

- characteristics of, 258
- creating
 - generic versus specific report structures, 259
 - requirements, determining, 258
 - workflows, incorporating into, 283-284
- designing
 - alternating row color, 274
 - sortable column headers, 276
- email, sending as, 281
- Excel (MS) documents, saving/sending as, 280
- list view layouts
 - printing, hiding elements before, 262
 - viewing, 262
- PDF, saving/sending as, 278
- printing, 262
- subsummary reports, 68, 266-267
 - summary fields, calculations, 268-270
 - viewing, 262
- summarized reports, subsummary parts, 266-268
- troubleshooting printed reports, 282

reproducing errors (script error handling), 434

Reset Account Password script step, 323

Resize To alignment tools (Arrange menu), 127

Resize To command (Arrange menu), 132

resizing layout objects, 118, 126, 129-130

Resource Center command, 29

restricted deletes, troubleshooting, 176

restricting custom function access, 375

results (scripts), 395

retrieving container field information, 50

Revert Record command deleting records from FileMaker Pro, 44 troubleshooting, 71

RGB functions, 363

RightValues functions, 366, 383

rollbacks, 418, 455

Round functions, 217

rows

- portal rows
 - creating/deleting, 55-56
 - highlighting, troubleshooting, 425
- report rows, alternating color, 274

runtime applications

- bind keys, adding/replacing files, 591
- creating via Developer Utilities (FileMaker Developer), 589, 592-593
- database extensions, 591

Runtime Name option (Solution Options), 591

runtime solutions, deploying, 589

S

SAT (Server Administration Tool), FileMaker Server, 603

Save As Excel scripts, 281

Save As option, 44

Save Records as PDF scripts, 280

saved scripts, 244

Find script steps, 245

Sort script steps, 246-247

saving

container field

information, 50

Excel documents, 69

PDF documents, 69

records, 44

reports

Excel, 280

PDF, 278

scheduling backups, 621, 634

scheduling feature (Admin Console), 620-621

schema imports (tables), 175-176

scope (variables)

global variables, 399

local variables, 398

scratch-building custom menus (user interfaces), 348

script completion, Allow User Abort scripts, 240

Script Debugger, 448-449

breakpoints, placing, 451

values, inspecting, 451

script errors (IWP application logs), 526

script libraries, building, 255

script logs, 290

-script-presort URL parameters, 571

-script.prefind URL parameters, 571

script results, 387

script steps

Add Account, 322

calculation formulas, 199

Change Password, 323

Commit Record/

Request, 292

Delete Account, 323

Enable Account, 323

If - Else If, 292

Open Record/Request, 292

Re-login, 323

Reset Account

Password, 323

-script URL parameters, 570

scripted imports, 506-507

ScriptMaker Interface, 233

Indicate Web Compatibility

check box, 533

Manage Scripts dialog, 232

Perform button, 233

Print button, 233

-scriptnames command, 569

scripts, 229

abstraction, 237

Allow User Abort

scripts, 240

branching, conditional

script steps, 247-248

button objects, associating

with, 239

buttons, building, 252-253

calculated replaces, 437

comments, 235, 431

conditional scripts, trou-

bleshooting, 254

creating, 230

custom dialogs, 250-251

data configuration

scripts, 241

Copy script steps, 242

Cut script steps, 242

Go to Field script

steps, 242

Paste script steps, 242

Set Field script steps,

242-243

data control scripts, 241

Copy script steps, 242

Cut script steps, 242

Go to Field script

steps, 242

Paste script steps, 242

Set Field script steps,
242-243

debugging, 448-449

inspecting values, 451

looping scripts, 451

placing breakpoints, 450

duties of, 229

editing, 233

error handling, 432

Get(LastError)

function, 433

reproducing errors, 434

Set Error Capture

scripts, 433

error management scripts,

239-240

Allow User Abort

scripts, 240

Set Error Capture

scripts, 240-241

error messages, trou-

bleshooting, 253

executing, 233

exiting, 236

exporting data, 519

full access privileges, 234

Go to Related Record

scripts, 420-421

disconnected table occur-

rences, jumping to, 422

versus searches, 438

hiding, 238

implementing, 232

importing, 233, 238

InitializeGlobals scripts, 255

IWP

committing records, 534

IWP execution tests, 535

startup/shutdown

scripts, 535

subscript calls, 535

unsupported script steps,

533-534

libraries, building, 255

loops, 248-249

calculated replaces, 437

conditional script

steps, 250

debugging, 451

exit conditions, 249-250

exiting, 250

- testing, 254
 - Manage Scripts dialog, 233
 - managing, 232-233
 - mode dependencies, table context, 441
 - modularizing, 393
 - Move/Resize Window, 414
 - moving, 232
 - multi-context locking, 293
 - multiple scripts, selecting, 233
 - parameters
 - modularizing scripts, 393
 - passing, 389
 - passing data between files, 394
 - passing multi-valued parameters, 390-392
 - retrieving, 389
 - specifying, 388
 - usage examples, 388, 393-394
 - post file conversion tasks, 463-465
 - PrintSetUp_landscape scripts, 255
 - PrintSetUp_portrait scripts, 255
 - record creation, slowness (performance), 438
 - record locking, trapping, 292, 295
 - record pointers, table context, 442
 - recursive, 401-402
 - reordering script steps, 234
 - results, 395
 - run privileges, security, 312
 - Save As Excel, 281
 - Save Records as PDF, 280
 - saved scripts, 244
 - Find script steps, 245
 - Sort script steps, 246-247
 - script steps, 241
 - Commit
 - Record/Request, 292
 - conditional script steps, 247-250
 - Copy script steps, 242
 - Cut script steps, 242
 - Else If script steps, 247-248
 - Else script steps, 247-248
 - End If script steps, 247-248
 - Find script steps, 245
 - Go to Field script steps, 242
 - Go to Layout script steps, 243
 - If - Else If, 292
 - If script steps, 247-248
 - My Set Field script steps, 254
 - Open Record/Request, 292
 - Paste script steps, 242
 - Set Field script steps, 242-243
 - Show Custom Dialog script steps, 250
 - Sort script steps, 246-247
 - ScriptMaker Interface
 - Manage Scripts dialog, 232
 - Perform button, 233
 - Print button, 233
 - ScriptTEMPLATE
 - scripts, 255
 - Send Mail, 281
 - Set Error Capture scripts, 240-241, 433
 - Set Variable script step, 398
 - ShutDown scripts, 255, 535
 - slowness (performance)
 - calculated replaces, 437
 - Go To Related Records scripts, 438
 - record creation, 438
 - Startup scripts, 255, 535
 - subscriptions, 230, 237
 - IWP, 535
 - table context, 441
 - templates, 236
 - ToggleAllStatusAreas scripts, 255
 - ToggleMultiUser scripts, 255
 - triggering, 251
 - unfinished scripts, troubleshooting, 253
 - user access, security, 302
 - user navigation scripts, 243-244
 - variables
 - dynamic file paths, 400
 - global variables, 399-400
 - local variables, 397-400
 - naming, 396
 - scope, 398-399
 - Set Variable script step, 398
 - viewing, 401
 - Web Viewer, controlling, 330-331
- ScriptTEMPLATE scripts, 255**
- searches**
- searching**
- dedicated find layouts, 423
 - Find mode layouts, 423
 - script-driven finds, 424
 - HTML-formatted search result stylesheets, building, 565-566
 - slowness in
 - (performance), 436
 - unindexed fields, 436
 - versus Go To Related Records scripts, 438
- securing Kiosk mode via Developer Utilities (FileMaker Developer), 595**
- security**
- access privileges, troubleshooting, 321
 - Custom Menus feature, 301
 - Edit Account dialog (Manage Accounts & Privileges dialog), 305
 - file-level access security, 316
 - external authentication, 319-320
 - file list filtering, 321
 - networks, 317
 - server
 - administration, 316
 - user authentication, 318

- FileMaker Pro 8, 21
- IWP, 525, 532
 - authenticating
 - accounts/privileges, 531
 - extended privileges, 529
 - passwords, 533
 - layouts, viewing, 113
 - multiple files, privilege sets, 322-323
 - opening/closing files, troubleshooting, 322
 - passwords, 302-303, 321
 - planning
 - aesthetics, 301
 - implementation, 301
 - matrixes, 299-301
 - script access, 302
 - user access, 302
 - user interfaces, 301
 - post-file conversion tasks, 461-462
 - primary concerns, 297
 - records, calculation
 - formulas, 200
 - risks, identifying, 298
 - user-level internal security
 - extended privileges, 314-316
 - privilege sets, 306-313
 - user accounts, 303-306
- Select All command (Edit menu), 126**
- selecting**
 - CWP technology, 575
 - fields in calculation
 - formulas, 201
 - FileMaker Server
 - configuration, 610
 - functions in calculation
 - formulas, 202
 - layout objects, 125-126
 - multiple scripts, 233
 - operators in calculation
 - formulas, 201
- Self function, 409-410**
- Send Backward command (Arrange menu), 131**
- Send Mail scripts, 281**
- Send to Back command (Arrange menu), 131**
- serial IDs, 290**
- serial key fields, creating, 91**
- Serial Number option (field auto-entry options), 90-91**
- server administration, 316**
- server hosting, 17**
- Server-IP section (URL), 552**
- server-side XSLT, CWP**
 - building HTML-formatted
 - search result stylesheets, 565-566
 - embedding query parameters in stylesheets, 566-567
 - stylesheet placement, 558
- servers. *See also* Filemaker Server**
 - connections, troubleshooting, 438-439
 - Windows servers, Mac plug-ins, 627
- sessions, 286**
 - database tracking, 288
 - globals, 287
 - login accounts, 287
 - troubleshooting, 294
 - IWP, 538
 - logins
 - testing, 294
 - troubleshooting, 294
 - managing, 572-573
 - session-specific
 - elements, 286
- Set Error Capture scripts, 240-241, 433**
- Set Field script steps, 242-243**
- Set Layout Order dialog, 114**
- Set Sliding/Printing dialog, 277**
 - Also Reduce the Size of the Enclosing Part option, 278
 - layout objects, nonprinting option, 278
 - Preview mode, viewing
 - effects in, 278
- Set Tab Order dialog, 138**
- Set Variable script step, 398**
- shadow fields, 489-490**
- sharing files in IWP, 530**
- sharing data between stylesheets, 567-568**
- Show All Records command (FileMaker Pro), 64**
- Show Custom Dialog script step, 250**
- Show in New Window option (Go to Related Record Options dialog), 421**
- Show Only Related Records option (Go to Related Record Options dialog), 421**
- Show Related Records From list (Portal Setup dialog), 166**
- Show/Hide Status Toolbar script step (IWP), 539**
- ShutDown scripts, 255, 535**
- simplifying complex formulas, 371-372**
- single computer FileMaker Server configuration, 608**
- single-user database deployment, 16**
- site profile options**
 - selecting with PHP Site Assistant, 582
 - specifying, 584-585
- Size palette, 118, 128-129**
- sizing**
 - layout parts, 118
 - IWP layout design restrictions, 537
 - table occurrences, 194
- skip URL parameters, 571**
- sliding layout objects, 277-278**
- slowness (performance)**
 - calculations, 437
 - general slowness, troubleshooting, 436

- scripts, 437-438
 - calculated replaces, 437
 - Go To Related Records scripts, 438
 - record creation, 438
- searching/sorting
 - slowness, 436
- software**
 - custom development software, 11
 - database software
 - advantages of, 9
 - as custom development software, 11
 - as off-the-shelf software, 10
 - functions of, 9-13
 - installing FileMaker Server maintenance tips, 633
 - off-the-shelf software, 10
 - updates, 27, 635
- Solution Options (Developer Utilities), 590**
 - Bindkey option, 591
 - Closing Splash Screen option, 592
 - Custom Image option, 592
 - Extension option, 591
 - Runtime Name option, 591
- solutions**
 - customizing via Developer Utilities (FileMaker Developer), 594
 - documenting, 457
 - kiosks, 589
 - runtime solutions, deploying, 589
- sort order, maintaining, 270**
- Sort script step, 246-247**
- sortfield command, 557**
- sortfield URL parameters, 571**
- sorting**
 - column headers (reports), 276
 - CWP URL search results, 557
 - IWP records, 544
 - multiple records, 66
 - multitiered sorting, 73
 - portals, 56
 - slowness in (performance), 436
 - unindexed fields, 436
 - value lists (fields), 67
- sortorder command, 557**
- sortorder URL parameters, 571**
- source tables, exporting data, 512**
- spacing in CWP URL, troubleshooting, 574**
- specific report structures, generic report structures versus, 259**
- Specify Calculation dialog, 85, 198-200**
 - Calculation Context section, 207-209
 - Calculation Result Is feature, 86
 - Data Type option, 204
 - Do Not Evaluate If All Referenced Fields Are Empty check box, 204-205
 - Field list feature, 85, 201
 - Formula text box feature, 86, 200
 - Function List, 202
 - Function list feature, 86
 - Number of Repetitions option, 204
 - Operators feature, 85, 201
- Specify Table dialog (Relationships Graph), 180**
- specifying external ODBC data sources, 485-486, 488**
- spell-checking in IWP, 533**
- SQL, 471, 492**
- SSL data transfer feature (FileMaker Server), 604**
- stacking orders, layout objects, 131**
- Standard Form layouts, 106**
- star joins, 157**
- Starter Solutions, 28**
- starting/stopping FileMaker Server, 616**
- StartUp scripts, 255**
- startup scripts, IWP, 535**
- Status Area. *See* Status toolbar**
- Status toolbar, 133-134**
 - Commit button, 539
 - hiding, 540
 - house icon, 527
 - Layout bar, 120
 - Mac OS X, customizing, 37-38
 - tool groups, 121-122
 - Windows, customizing, 38-39
- Storage Options dialog**
 - Global Storage option, 205
 - Indexing option, 206-207
- storing fields**
 - globals, 96
 - repeating fields, 97
- strings (text)**
 - case altering functions, 214
 - concatenation, 212
 - defining, 212
 - Exact functions, 213
 - interrogating, 212
 - Length functions, 212
 - PatternCount functions, 212
 - Position functions, 213
 - Substitute functions, 214
 - text parsing functions, 215
 - Trim functions, 213-214
 - WordCount functions, 213
- structured data elements, passing multi-valued script parameters, 392**
- stylehref URL parameters, 571**
- styletype URL parameters, 571**

stylesheets

- HTML-formatted search result stylesheets, building, 565-566
- query parameters, embedding in, 566-567
- sharing data, 567-568
- WPE directories, placement in, 558

styling text in FileMaker Pro, 73**subscripts, 230, 237, 535****Substitute functions, 214****subsummary reports, 68, 116, 119**

- multi-column layouts, 114
- rearranging, 267
- summary fields, calculations, 268-270
- viewing, 262

summarized reports, sub-summary parts, 266-268**summary fields, 49, 87-89**

- subsummary reports, calculations, 268-270

T

Tab character transformations (exporting data), 514**Tab Control Setup dialog (Status Toolbar), 133-134****tab order, fields, 138****tab-separated text format (exporting data), 515****table contexts, 553**

- troubleshooting, 439
- calculations, 440
- importing/exporting records, 440
- layouts, 440
- mode dependencies, 441
- record pointers, 442
- scripts, 441
- value lists, 441

Table menu (Manage Database dialog), Table Name box, 161**Table Name box (Table menu), 161****table occurrences, 19, 163, 180**

- Go to Related Record scripts, 422
- layouts, 108-109
- sizing, 194

Table View layouts, 33, 71, 106, 271**tables, 30**

- creating, 78
- CWP URL searches, 553
- definitions
 - copying/pasting, 176
 - importing between files, 175
- deleting, 79
- employee tables, example of, 12
- external tables, adding to Relationships Graphs, 191
- fields, naming conventions, 79-81
- join tables, naming, 428
- layouts, viewing as, 112
- multifile table systems, 187-188
- multitable systems
 - adding to, 161
 - building first tables in, 160
 - portals, 165-167, 169
 - naming conventions, 77-78
- portals
 - creating/deleting rows, 55-56
 - mechanics of, 55
 - sorting, 56
 - schema imports, 175-176

tabs, applying to text, 73**tar archives, Mac plug-ins, 627****targets, 404****templates**

- scripts, 236
- Web Viewer, 329

testing

- Boolean tests, 355
- conditional tests, 354
- logins, 294
- loops, 254

text

- formatting functions, troubleshooting in nontext calculations, 382
- misspellings in fields, 45
- styling, 73
- tabs, applying, 73

text arrays, passing multi-valued script parameters, 390**text fields, 49, 83, 93****text files, importing, 502-503****text formatting functions, 362**

- RGB functions, 363
- TextColor functions, 363
- TextColorRemove functions, 364
- TextFont functions, 363
- TextFontRemove functions, 364
- TextFormatRemove functions, 364
- TextSize functions, 363
- TextSizeRemove functions, 364
- TextStyleAdd functions, 363
- TextStyleRemove functions, 363

text functions, 212

- case altering functions, 214
- Exact functions, 213
- Length functions, 212
- PatternCount functions, 212
- Position functions, 213
- Substitute functions, 214
- text parsing functions, 215
- Trim functions, 213-214
- WordCount functions, 213

text parsing functions, 215**text strings**

- case altering functions, 214
- concatenation, 212
- defining, 212

- Exact functions, 213
- interrogating, 212
- Length functions, 212
- PatternCount functions, 212
- Position functions, 213
- Substitute functions, 214
- text parsing functions, 215
- Trim functions, 213-214
- WordCount functions, 213
- TextColor functions, 363**
- TextColorRemove functions, 364**
- TextFont functions, 363**
- TextFontRemove functions, 364**
- TextFormatRemove functions, 364**
- TextSize functions, 363**
- TextSizeRemove functions, 364**
- TextStyleAdd functions, 363**
- TextStyleRemove functions, 363**
- three computer FileMaker Server configuration, 609**
- thumbnails, importing, 504-505**
- time fields, 49, 83**
- Time functions, 218-220**
- timers, 408**
- timestamp fields, 49. See also data fields**
 - Creation Timestamp fields, 93
 - Modification Timestamp fields, 93
 - time fields, 84
- TimeStamp functions, 219-220**
- Title Footers, 117**
- Title Headers, 116**
- ToggleAllStatusAreas scripts, 255**
- ToggleMultiUser scripts, 255**
- token URL parameters, 571**
- tokens, sharing data between stylesheets, 567-568**
- tool groups(Status toolbar), 121-122**
- tool palettes (multi-window interfaces), 415**
- Tools menu, Data Viewer**
 - Debug Scripts feature, 451
 - expressions, defining, 451
- tooltips, 140**
- Top to Bottom alignment option (Align command), 132**
- tracking database sessions, 288**
- Trailing Grand Summaries, 114-116**
- transactions (databases), 288**
- transferring data, SSL data transfers, 604**
- trapping**
 - errors, troubleshooting, 295
 - record locking in scripts, 292
- triggers, 403-404**
 - event handlers, 405
 - for layouts, 406
 - for objects, 407
 - for scripts, 251
 - functions, Self function, 409-410
 - targets, 404
- Trim functions, 213-214**
- TrimChar functions, 379**
- troubleshooting**
 - audit trails, 295
 - conditional scripts, 254
 - connectivity, server connections, 438-439
 - context, 439
 - layout dependencies, 439
 - table context, 439-442
 - CWP
 - spacing in URL, 574
 - URL requests, 573
 - data imports, 508-509
 - data integrity, 70
 - data loss, 70
 - data types in calculation fields, 226
 - error trapping, 295
 - field names in calculation formulas, 225-226
 - fields
 - copying/pasting, 141
 - field validation, 101
 - mismatched calculations, 100
 - mismatched data types, 100
 - naming conventions, 101
 - file conversion process, 468-469
 - file reference errors, 600
 - FileMaker Server
 - access, 573
 - firewalls, CWP URL requests, 573
 - globals, 294, 442-443
 - IWP, ending sessions, 544
 - LeftValue functions, 383
 - Let functions, 383
 - logins, 294
 - MiddleValues functions, 383
 - multiple-match relationships, 193
 - nonequijoins, 192-193
 - performance, 435
 - calculation slowness, 437
 - general slowness, 436
 - script slowness, 437-438
 - searching/sorting slowness, 436
 - plug-ins, 600
 - pop-up windows, 424
 - portal rows,
 - highlighting, 425
 - proactive troubleshooting
 - debugging calculations, 434
 - debugging scripts, 448-451
 - planning for failure, 431
 - scripts, error handling, 432, 434
 - writing code, 428-431

- records, displaying in layouts, 141
- related records, 254
- repeating portals, 176
- reports, printed reports, 282
- restricted deletes, 176
- Revert Record command (FileMaker Pro), 71
- RightValues functions, 383
- script error messages, 253
- script steps, 254
- security
 - access privileges, 321
 - converted passwords, 321
 - forgotten Admin passwords, 321
 - opening/closing files, 322
- SQL, 492
- text formatting functions, nontext calculations, 382
- unfinished scripts, 253
- ValueCount functions, 383
- window construction, lost found sets, 424
- WPE access, 573

Truncate functions, 217

two computer FileMaker Server configuration, 609

U

UI files, 339-340

Undo command, 44

unfinished scripts, troubleshooting, 253

unindexable fields, 100, 436

unsupported script steps (IWP)

- error capturing, 534
- identifying, 533

updates, software updates for FileMaker Server maintenance, 635

updating

- custom functions, 373
- plug-ins via FileMaker Server, 626-632
- records with imported data, 497-498

upgrading FileMaker

- files, converting, 454-455
 - multifile relational solutions, 456-457
- post-conversion tasks, 460-468
- security, 461-462
 - preconversion tasks, 457-460
 - single-file solutions, 455-456
 - troubleshooting, 468-469
- web-enabled databases, converting, 469-470

UPS (uninterruptible power supply), 443

URL

- components of, 552
- CWP
 - exact match searches, 554-555
 - multiple criteria searches, 556
 - multiple find requests, 556
 - numerical comparison searches, 555
 - specific record searches, 554
 - specifying search result sort order, 557
 - table searches, 553
 - troubleshooting, 573

Grammar section, 552

parameters

- db, 570
- encoding, 570
- field, 570
- fieldname, 570
- Fieldname.op, 570
- grammar, 570
- lay, 570
- lay.response, 570
- lop, 570
- max, 570
- modid, 570
- recid, 570
- script, 570
- script-presort, 571
- script.prefind, 571

- skip, 571
- sortfield, 571
- sortorder, 571
- stylehref, 571
- styletype, 571
- token, 571

Port section, 552

Protocol section, 552

query strings, 552-553

Server-IP section, 552

spacing in, troubleshooting, 574

XSLT URL formats, 566

user accounts

- access privileges, troubleshooting, 321
- privilege sets, reviewing, 305
- user-level internal security, 303
 - Admin accounts, 304
 - automatic logins, 304
 - default accounts, 304
 - editing, 304-306
 - [Guest] accounts, 304
 - managing, 304, 306

user authentication, security, 318

user fields, naming, 429

user IDs, securing, 302-303

user interfaces

- building, 336-339

- custom menus

- components of, 341

- editing, 345

- FileMaker control

- of, 342

- managing, 342-344

- menu sets interface, 342-344

- menu sets, assigning, 346

- menu sets, loading/

- activating, 346-347

- scratch-building, 348

- uses of, 341

- dedicated find layouts, 423

- Find mode layouts, 423

- script-driven finds, 424

- design guidelines, 349-350

- multi-window interfaces
 - dialog windows, 416-419
 - tool/function palettes, 415
 - uses of, 415
- native user interface, 336
- prototype layouts/
 - menus, 335
- security plans, 301
- UI files, 339-340

user navigation scripts, 243-244

- user-level internal security**
 - extended privileges
 - custom privileges, 315-316
 - default privileges, 314-315
 - privilege sets, 306-307
 - conditional
 - privileges, 310
 - data access, controlling, 308-309
 - data validation warning
 - override privileges, 313
 - export privileges, 313
 - FileMaker Server
 - disconnects, 313
 - layout use/development, 310-311
 - menu commands
 - access, 313
 - passwords, setting, 313
 - printing privileges, 313
 - run script privileges, 312
 - value list access, 311-312
 - user accounts, 303
 - Admin accounts, 304
 - automatic logins, 304
 - default accounts, 304
 - editing, 304-306
 - [Guest] accounts, 304
 - managing, 304-306

V

- validating data**
 - fields, 94-96
 - calculation formulas, 200
 - condition failures, 95
 - troubleshooting, 101
 - FileMaker Pro, 52
 - override privileges, 313

Validation dialog, 94-95

- Value from Last Visited Record option (field auto-entry options), 91**

value indexes, 98-99

- value lists**
 - access privileges, 311-312
 - creating for many-to-many relationships, 173-174
 - IWP, 533
 - table context, 441

- value lists (fields)**
 - editing, 47-48
 - “other” values, 48
 - sorting, 67

ValueCount functions, 365, 383

ValueListItems functions, 225

- variables (scripts)**
 - dynamic file paths, 400
 - global variables
 - globally stored fields
 - versus, 399
 - naming, 399
 - scope, 399
 - user session data, 399
 - uses for, 400
 - local variables, 397
 - scope, 398
 - uses for, 400
 - naming, 396
 - scope
 - global variables, 399
 - local variables, 398
 - Set Variable script step, 398
 - viewing, 401

- verifying processes, script logs, 290**

View As options, IWP layout design restrictions, 536

- view command, 569**

View setting (Custom Privileges dialog), 309

- viewing**
 - layouts
 - forms view, 111
 - lists view, 112, 262
 - restricting user access, 113
 - tables view, 112
 - plug-in names/version numbers, 630-631
 - related child data in multiple tables, 165-167
 - subsummary reports, 262

views, 31

Views tab (Layout Setup dialog), 113

volume licenses, 26

W

Web pages, linking to IWP files, 541

Web publishing

- Access via Instant Web Publishing extended privilege, 314-315
- errors (IWP application logs), 526

Web Publishing Engine, 608

web servers, 608

Web Viewer

- configuring, 325, 328
- controls, adding, 330-331
- current page, identifying, 331-332
- files, viewing, 332
- templates, 329

websites, modifying xslt-template files, 564-565

WeekEndingFriday functions, 376**wildcards**

IP addresses, 525
searches, 62

WindowNames functions, 225, 413**windows**

building, 424
managing, 413
troubleshooting, 424

Windows Operating System

ODBC administration, 476
data source names, 477
drivers, 478
DSN, configuring, 483-484
pop-up windows, troubleshooting, 424
Status toolbar, customizing, 38-39

Windows servers, Mac plug-ins, 627**windowshades, table occurrences, 194****WKS file format (exporting data), 516****Word (MS), formatting files as DDR, 446****word indexes, 98****word separators, 213****WordCount functions, 213****workflows, incorporating into reports, 283-284****WPAC (Web Publishing Administration Console), 527****WPE (Web Publishing Engine)**

accessing, 573
CWP URL searches
exact match searches, 554-555
multiple criteria searches, 556
numerical comparison searches, 555

specific record searches, 554
table searches, 553
multiple find requests, 556
search results, specifying sort order, 557
URL, spaces in, 574
XML publishing, 549-553, 557
XSLT, building HTML-formatted search result stylesheets, 565-566
stylesheet placement, 558

writing

calculation formulas, 198, 200-201
legibility, 202
fields, selecting, 201
functions, selecting, 202
operators, selecting, 201
tips, 226-228
code
comments, 430-431
naming database files, 428
naming fields, 429
naming functions, 430
naming layouts, 429
naming parameters, 430
plug-ins, 596

X-Y-Z**XLST Site Assistant, modifying xslt-template-files, 564-565****XML (eXtensible Markup Language)**

CWP, 549-553, 557
-find command, 554-555
-lop command, 556
processing instructions, embedding query parameters in stylesheets, 566-567
processing instructions, 567
query strings, 567
-sortfield command, 557
-sortorder command, 557

XML file format (exporting data), 515**XML Web Publishing, Access via XML Web Publishing extended privilege, 315****XMpl_Add function, 598****XMpl_Append function, 598****XMpl_NumToWords function, 598****XMpl_StartScript function, 598****XMpl_UserFormatNumber function, 598****XSL extensions, CWP, 572****XSLT****CWP**

building HTML-formatted search result stylesheets, 565-566
embedding query parameters in stylesheets, 566-567
stylesheet placement, 558
XSL extensions, 572
XSLT processors, 571
processors, CWP, 571
query strings, 566
URL formats, 566

XSLT Site Assistant, 559**XSLT Web Publishing, Access via XSLT Web Publishing extended privilege, 315****xslt-template-files, modifying, 564-565**