

THE OFFICIAL PENDRAGON FORMS™ FOR PALM OS®



Transform Palm OS devices into mobile data-collection tools
Streamline data collection with customized Pendragon forms
Synchronize Pendragon Forms with your existing databases

**Debra Sancho
and Ivan Phillips**

Pendragon Software Corporation

RPX-1012
Page 1 of 444



Pendragon
evaluation
and more

SPECIAL
Upgrade
evaluation copy
for only **\$99.95**
(regularly **\$149**)

See inside for details

The Official Pendragon Forms™ for Palm OS Starter Kit

The Official Pendragon Forms™ for Palm OS Starter Kit

Debra Sancho and Ivan Phillips

M&T Books
An Imprint of IDG Books Worldwide, Inc.



Foster City, CA ♦ Chicago, IL ♦ Indianapolis, IN ♦ New York, NY

The Official Pendragon Forms™ for Palm OS Starter Kit

Published by
M&T Books

An imprint of IDG Books Worldwide, Inc.
919 E. Hillsdale Blvd., Suite 400
Foster City, CA 94404

www.idgbooks.com (IDG Books Worldwide Web site)

Copyright © 2000 IDG Books Worldwide, Inc. All rights reserved. No part of this book, including interior design, cover design, and icons, may be reproduced or transmitted in any form, by any means (electronic, photocopying, recording, or otherwise) without the prior written permission of the publisher.

HotSync, Palm OS, and Palm Computing are registered trademarks, and Palm is a trademark of Palm Computing, Inc. or its subsidiaries.

Library of Congress Catalog Card Number: 99-068577
ISBN: 0-7645-4651-1

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

1B/QR/RS/ZZ/FC

Distributed in the United States by IDG Books
Worldwide, Inc.

Distributed by CDG Books Canada Inc. for Canada; by Transworld Publishers Limited in the United Kingdom; by IDG Norge Books for Norway; by IDG Sweden Books for Sweden; by IDG Books Australia Publishing Corporation Pty. Ltd. for Australia and New Zealand; by TransQuest Publishers Pte Ltd. for Singapore, Malaysia, Thailand, Indonesia, and Hong Kong; by Gotop Information Inc. for Taiwan; by ICG Muse, Inc. for Japan; by Intersoft for South Africa; by Eyrolles for France; by International Thomson Publishing for Germany, Austria and Switzerland; by Distribuidora Cuspide for Argentina; by I.R. International for Brazil; by Galileo Libros for Chile; by Ediciones ZETA S.C.R. Ltda. for Peru; by WS

Computer Publishing Corporation, Inc., for the Philippines; by Contemporanea de Ediciones for Venezuela; by Express Computer Distributors for the Caribbean and West Indies; by Micronesia Media Distributor, Inc. for Micronesia; by Chips Computadoras S.A. de C.V. for Mexico; by Editorial Norma de Panama S.A. for Panama; by American Bookshops for Finland.

For general information on IDG Books Worldwide's books in the U.S., please call our Consumer Customer Service department at 800-762-2974. For reseller information, including discounts and premium sales, please call our Reseller Customer Service department at 800-434-3422.

For information on where to purchase IDG Books Worldwide's books outside the U.S., please contact our International Sales department at 317-596-5530 or fax 317-596-5692.

For consumer information on foreign language translations, please contact our Customer Service department at 800-434-3422, fax 317-596-5692, or e-mail rights@idgbooks.com.

For information on licensing foreign or domestic rights, please phone +1-650-655-3109.

For sales inquiries and special prices for bulk quantities, please contact our Sales department at 650-655-3200 or write to the address above.

For information on using IDG Books Worldwide's books in the classroom or for ordering examination copies, please contact our Educational Sales department at 800-434-2086 or fax 317-596-5499.

For press review copies, author interviews, or other publicity information, please contact our Public Relations department at 650-655-3000 or fax 650-655-3299.

For authorization to photocopy items for corporate, personal, or educational use, please contact Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, or fax 978-750-4470.

LIMIT OF LIABILITY/DISCLAIMER OF WARRANTY: THE PUBLISHER AND AUTHOR HAVE USED THEIR BEST EFFORTS IN PREPARING THIS BOOK. THE PUBLISHER AND AUTHOR MAKE NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS BOOK AND SPECIFICALLY DISCLAIM ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. THERE ARE NO WARRANTIES WHICH EXTEND BEYOND THE DESCRIPTIONS CONTAINED IN THIS PARAGRAPH. NO WARRANTY MAY BE CREATED OR EXTENDED BY SALES REPRESENTATIVES OR WRITTEN SALES MATERIALS. THE ACCURACY AND COMPLETENESS OF THE INFORMATION PROVIDED HEREIN AND THE OPINIONS STATED HEREIN ARE NOT GUARANTEED OR WARRANTED TO PRODUCE ANY PARTICULAR RESULTS, AND THE ADVICE AND STRATEGIES CONTAINED HEREIN MAY NOT BE SUITABLE FOR EVERY INDIVIDUAL. NEITHER THE PUBLISHER NOR AUTHOR SHALL BE LIABLE FOR ANY LOSS OF PROFIT OR ANY OTHER COMMERCIAL DAMAGES, INCLUDING BUT NOT LIMITED TO SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR OTHER DAMAGES.

Trademarks: All brand names and product names used in this book are trade names, service marks, trademarks, or registered trademarks of their respective owners. IDG Books Worldwide is not associated with any product or vendor mentioned in this book.



is a registered trademark or trademark under exclusive license to IDG Books Worldwide, Inc. from International Data Group, Inc. in the United States and/or other countries.



is a trademark of IDG Books Worldwide, Inc.

ABOUT IDG BOOKS WORLDWIDE

...to the world of IDG Books Worldwide.

IDG Books Worldwide, Inc., is a subsidiary of International Data Group, the world's largest publisher of computer-related information and the leading global provider of information services on information technology. IDG was founded more than 30 years ago by Patrick J. McGovern and now employs more than 9,000 people worldwide. IDG publishes more than 290 computer publications in over 75 countries. More than 90 million people read one or more IDG publications each month.

Launched in 1990, IDG Books Worldwide is today the #1 publisher of best-selling computer books in the United States. We are proud to have received eight awards from the Computer Press Association in recognition of editorial excellence and three from Computer Currents' First Annual Readers' Choice Awards. Our best-selling ...For Dummies® series has more than 50 million copies in print with translations in 31 languages. IDG Books Worldwide, through a joint venture with IDG's Hi-Tech Beijing, became the first U.S. publisher to publish a computer book in the People's Republic of China. In record time, IDG Books Worldwide has become the first choice for millions of readers around the world who want to learn how to better manage their businesses.

Our mission is simple: Every one of our books is designed to bring extra value and skill-building instructions to the reader. Our books are written by experts who understand and care about our readers. The knowledge base of our editorial staff comes from years of experience in publishing, education, and journalism — experience we use to produce books to carry us into the new millennium. In short, we care about books, so we attract the best people. We devote special attention to details such as audience, interior design, use of icons, and illustrations. And because we use an efficient process of authoring, editing, and desktop publishing our books electronically, we can spend more time ensuring superior content and less time on the technicalities of making books.

You can count on our commitment to deliver high-quality books at competitive prices on topics you want to read about. At IDG Books Worldwide, we continue in the IDG tradition of delivering quality for more than 30 years. You'll find no better book on a subject than one from IDG Books Worldwide.



John Kilcullen
Chairman and CEO
IDG Books Worldwide, Inc.



Steven Berkowitz
President and Publisher
IDG Books Worldwide, Inc.



Eighth Annual
Computer Press
Awards 1992



Ninth Annual
Computer Press
Awards 1993



Tenth Annual
Computer Press
Awards 1994



Eleventh Annual
Computer Press
Awards 1995

IDG is the world's leading IT media, research and exposition company. Founded in 1964, IDG had 1997 revenues of \$2.05 billion and has more than 9,000 employees worldwide. IDG offers the widest range of media options that reach IT buyers in 75 countries representing 95% of worldwide IT spending. IDG's diverse product and services portfolio spans six key areas including print publishing, online publishing, expositions and conferences, market research, education and training, and global marketing services. More than 90 million people read one or more of IDG's 290 magazines and newspapers, including IDG's leading global brands — Computerworld, PC World, Network World, Macworld and the Channel World family of publications. IDG Books Worldwide is one of the fastest-growing computer book publishers in the world, with more than 700 titles in 36 languages. The "...For Dummies®" series alone has more than 50 million copies in print. IDG offers online users the largest network of technology-specific Web sites around the world through IDG.net (<http://www.idg.net>), which comprises more than 225 targeted Web sites in 55 countries worldwide. International Data Corporation (IDC) is the world's largest provider of information technology data, analysis and consulting, with research centers in over 41 countries and more than 400 research analysts worldwide. IDG World Expo is a leading producer of more than 168 globally branded conferences and expositions in 35 countries including E3 (Electronic Entertainment Expo), Macworld Expo, ComNet, Windows World Expo, ICE (Internet Commerce Expo), Agenda, DEMO, and Spotlight. IDG's training subsidiary, ExecuTrain, is the world's largest computer training company, with more than 230 locations worldwide and 785 training courses. IDG Marketing Services helps industry-leading IT companies build international brand recognition by developing global integrated marketing programs via IDG's print, online and exposition products worldwide. Further information about the company can be found at www.idg.com. 1/24/99

Credits

ACQUISITIONS EDITORS

Ann C. Lush
Amy Barkat

DEVELOPMENT EDITOR

Eric Newman

TECHNICAL EDITOR

Ivan Phillips

COPY EDITORS

Robert Campbell
Mildred Sanchez

PROJECT COORDINATORS

Linda Marousek
Joe Shines

GRAPHICS AND PRODUCTION SPECIALISTS

Mario Amador
Stephanie Hollier
Jude Levinson
Ramses Ramirez
Dina F Quan

QUALITY CONTROL SPECIALIST

Chris Weisbart

BOOK DESIGNER

Jim Donohue

ILLUSTRATOR

Heather Hudson

PROOFREADING AND INDEXING

York Production Services

COVER DESIGN

Deborah Reinerio

About the Authors

Debra Sancho is a vice president with Pendragon Software and has been with the company for two years. Debra has a master's degree in Corporate Training from Northwestern University and prior to joining Pendragon Software was an instructional designer at United Airlines for five years. Debra has written much of the technical documentation for Pendragon Software products.

Ivan Phillips is president of Pendragon Software and the lead developer for Pendragon Forms. He has a Ph.D. in theoretical physics from Northwestern University. Ivan has been developing software for more than 15 years and has cowritten two books on the Java programming language.

*For our parents,
Ann & Gerry
and
Norman & Sandra*

Preface

The Palm organizer started out as a Personal Digital Assistant (or PDA) used by individuals for addresses and appointments. The widespread appeal of the device has revolutionized the handheld computer industry and has made mobile computing a reality.

In November 1996, only a few months after the release of the first Palm Computing PDA, Pendragon Software released the first version of Pendragon Forms (then called PilotForms 1.0). Since then, Pendragon Forms has been enhanced several times to make the software more flexible and more capable. With each new version, the software manual has grown, but there has been an increasing need to provide more in-depth information and advice. When we were approached with the idea to create a complete “starter kit,” we jumped at the opportunity.

This book goes beyond the standard reference guide that ships with the software to provide tips, techniques, and a wealth of examples. All of the examples and a 30-day evaluation version of the software are provided on the accompanying CD-ROM.

Beginning and intermediate-level topics are covered in this book. Starting with the basics of creating a form to run on the Palm organizer, the book progresses to more advanced topics, such as setting up remote access and network synchronization. The most advanced techniques that relate to the internal operation of the software are not included here, though they are documented in help files on the CD-ROM.

How to Use This Book

If you are wondering how to move from a pen-and-paper data collection environment to using Palm organizers for a mobile workforce, then this book is for you. *The Official Pendragon Forms for Palm Starter Kit* shows you how to go beyond the built-in applications on the handheld to create forms that are customized for the type of data that you record every day in your job.

While writing this book, we have assumed that you own a Palm organizer and are familiar with handheld activities such as performing a HotSync data transfer, using the Graffiti writing system, and using the built-in Address Book application.

Your job can be in any industry. We have assumed that you are comfortable using computers but are not necessarily a programmer.

You can use this book to quickly master the basics of creating a custom form for the Palm organizer. Once you've learned how easy it is to create a form, you can use this book as a reference in order to optimize the data entry experience that the handheld user has while using Pendragon Forms.

How This Book Is Organized

The Official Pendragon Forms for Palm Starter Kit contains seven parts.

Part I: Creating Customized Forms for the Palm Organizer

Chapters in Part I look at why the Palm Computing platform has been so successful, and why Palm computers are an excellent choice for deploying mobile applications. You will see how Pendragon Forms provides an easy way to create your own custom data-entry forms for the Palm organizer, and you will consider the alternatives to using Pendragon Forms. Part I also shows you the basics of creating a form in Pendragon Forms and provides you with tips and techniques for using Pendragon Forms on the handheld.

Part II: Form Design Techniques

Chapters in Part II consider some of the issues that enable you to make the most of the limited resources available on a Palm computer. We review the 21 different field types that you can use when designing a form and discuss advanced field properties and advanced form properties that give you control of individual fields and forms. Part II also looks at the synchronization rules that determine how the Pendragon Forms conduit functions during the HotSync process to send form designs to the handheld and to send data from handheld to PC and from PC to handheld.

Part III: Managing Data and Form Designs

Chapters in Part III look at your options for viewing data on the PC and creating reports in Microsoft Word and Microsoft Access. Part III also outlines how to back up the Pendragon Forms database and looks at how to import and export form designs and Lookup Lists.

Part IV: Creating Specialized Forms

Chapters in Part IV discuss scripts, which bring a whole new realm of functionality to a Pendragon form – with scripts you can perform calculations on the handheld or program buttons to perform actions when the handheld user taps the button. Part IV also reviews the use of bar codes with Pendragon Forms. (Bar code scanning hardware is required.)

Part V: Using Pendragon Forms with Existing Databases

Chapters in Part V show you how to create a Pendragon form based on a database table in an existing Microsoft Access database. You can link the form to the external database table so that when you perform a HotSync data transfer, data from the Palm organizer goes directly to your database. Part V also covers linking Pendragon Forms to ODBC databases such as Foxpro, dBase, SQL Server, and Oracle.

Part VI: Working in a Multi-user Environment

Part VI looks at the issues involved in deploying Pendragon Forms applications to multiple handhelds. The benefits and limitations of the HotSync Manager, Network HotSync, and WaveSync synchronization server are discussed. Another way to deploy a multiuser application is through the use of a wireless Palm VII solution with Pendragon Forms, and Part VI outlines how scripts can be used to transmit a record to a Web site.

Appendixes

Finally, three appendixes provide troubleshooting tips, information on scripting syntax, and a guide to the contents of the CD-ROM.

Conventions

The following conventions are used in this book to make the material easier to understand:

- ◆ Key combinations such as Ctrl+P are joined by plus signs.
- ◆ A notation containing arrows is used for progressive menu selections. For example, Start → Programs → Pendragon Forms means click the Start button in Windows, then select the Programs menu, and then select the Pendragon Forms menu choice.
- ◆ *Italics* are used for emphasis.
- ◆ Monotype text indicates code.
- ◆ A notation that contains a forward slash is used for Graffiti shortcuts on the handheld. For example, /W means enter a forward slash in Graffiti (by drawing a line that moves diagonally upward), followed by the Graffiti letter W.

The icons used for the special features are shown next.



The On the CD-ROM icon refers to an evaluation program or sample form that can be found on the CD-ROM at the back of this book.



The Tip icon gives you insider information on using various features of Pendragon Forms.



The Cross-Reference icon refers to a related topic in another chapter of the book, or sometimes elsewhere in the same chapter.



The Note icon is used to inform you of special cases or exceptions to the normal way that a feature of Pendragon Forms works.



The Caution icon warns you about possible problems that you may encounter when using Pendragon Forms.

What Is a Sidebar?

Topics that appear in sidebars provide in-depth hints on technical issues.

Acknowledgments

This book would not have been possible without the help of a great team at IDG Books. We thank Joe Wikert for the idea for the Official Starter Kit, and Ann Lush and Amy Barkat for helping to make the vision a reality. We thank Eric Newman, our development editor, for taking what must have looked like pieces of a giant puzzle and making it whole. Thanks also to Robert Campbell and Mildred Sanchez, our copy editors, and to everyone who worked on the many graphics that brought this book to life.

Thanks to everyone in the Pendragon Forms Launch Control Team for another successful launch, and thanks to our Pendragon Forms customers who find new uses for forms on the Palm organizer every day.

Contents at a Glance

	Preface	ix
	Acknowledgments	xiii
Part I	Creating Customized Forms for the Palm Organizer	
Chapter 1	What Is Pendragon Forms?	3
Chapter 2	Creating a Form	13
Chapter 3	Entering Data on the Palm Organizer	33
Part II	Form Design Techniques	
Chapter 4	Planning a Form Design	57
Chapter 5	Field Types	67
Chapter 6	Advanced Field Properties	117
Chapter 7	Synchronization Rules	137
Chapter 8	Advanced Form Properties	153
Part III	Managing Data and Form Designs	
Chapter 9	Managing Data on the PC	177
Chapter 10	Managing Form Designs	201
Part IV	Creating Specialized Forms	
Chapter 11	Using Scripts	215
Chapter 12	Using Bar Codes	265
Part V	Using Pendragon Forms with Existing Databases	
Chapter 13	Linking to an External Access Database	285
Chapter 14	Linking to an ODBC Database	317
Part VI	Working in a Multi-User Environment	
Chapter 15	Planning a Multi-User Installation	333
Chapter 16	Working with the Palm VII	347

Appendix A: Troubleshooting Tips	355
Appendix B: Scripting Syntax	375
Appendix C: What's on the CD-ROM?	379
Glossary	393
Index	399
End-User License Agreement	427
CD-ROM Installation Instructions	432

Contents

Preface	ix
Acknowledgments	xiii

Part I **Creating Customized Forms for the Palm Organizer**

Chapter 1	What Is Pendragon Forms?	3
	The Palm Computing Platform	3
	Size and Battery Life	3
	Graffiti Handwriting Recognition	4
	HotSync Technology	4
	Types of Palm Devices	4
	What Is Pendragon Forms?	6
	How Does Pendragon Forms Work?	6
	Deciding if Pendragon Forms Is the Right Development Tool .	7
	How to Use This Book and CD-ROM	9
	Getting Started: Installing Pendragon Forms	10
Chapter 2	Creating a Form	13
	Designing a Form	13
	Editing a Form Design	19
	Freezing a Form Design	20
	Sending a Form to the Handheld	24
	Entering Records on the Handheld	25
	Using Field View and Record View	26
	Uploading and Viewing Data on the PC	28
	Bi-directional Synchronization	30
	Making Changes to a Frozen Form	31
	Copying a Form Design	31
	Printing a Form Design	32
Chapter 3	Entering Data on the Palm Organizer	33
	Entering New Records on the Handheld	33
	Working in Field View	34
	Using AutoNavigate	35
	Working in Record View	37
	Jumping from Record View to Field View and Back	39
	Changing the Default View for New Records	40
	Reviewing Records on the Handheld	40
	Filtering Records	44
	Reviewing Filtered Records	44

Filtering Records with an SPT 1500 Handheld	45
Sorting Records	45
Sending Records to the PC	46
Identifying New and Changed Records	47
Identifying Problem Records That Did Not Upload to the PC	48
Deleting Records and Forms from the Handheld	49
Manually Deleting Forms and Records from the Handheld	49
Limiting Handheld Access Rights	50
Printing from the Handheld	52
Infrared Printing	52
Serial Port Printing	52
Printing a Record	52
Printer Output	53

PART II Form Design Techniques

Chapter 4	Planning a Form Design	57
	Planning a Form	57
	Form Usability	57
	Tips for Designing Forms	58
	Performance	58
	How Performance Depends on Your Data	59
	Estimating Memory Usage	59
	Reducing the Amount of Data on the Handheld	62
	How Often Should You Synchronize?	63
	Synchronization Speed	64
	Modem HotSync Transfers	64
	Network HotSync Transfers	64
	Server Synchronization	65
	Tips for Improving Synchronization Performance	66
Chapter 5	Field Types	67
	Pendragon Forms' 21 Field Types	67
	Freeform Text Field	68
	Numeric Field	70
	Currency Field	72
	Yes or No Checkbox Field	73
	Option 1 of 5 Field	74
	Popup List Field	76
	Multi-Selection List Field	78
	Lookup List and Exclusive Lookup List Fields	80
	Cascading Lookup	85
	Lookup to Another Form	88
	Date & Time Field	91
	Date Only Field	93

	Time Field	94
	Time Checkbox	95
	Section Field	97
	Jump Popup Field	99
	Subform List Field	100
	Single Subform Field	107
	Signature Field	110
	Button Field	112
Chapter 6	Advanced Field Properties	117
	Column Name	119
	Default Value	120
	Hidden	123
	Read-Only	124
	Autodefaut	124
	Display Key	125
	Required	126
	Primary Key	128
	Selecting a Primary Key	128
	Working with Primary Keys on the Handheld	129
	Do Not Upload to PC	131
	Non-Printing	132
	Max and Min	132
	Max Length	133
	Lookup List	134
	Pattern	135
Chapter 7	Synchronization Rules	137
	The Pendragon Forms Conduit	137
	How Does the HotSync Manager Work?	138
	The Synchronization Process	139
	The User List	140
	Adding Handheld Users	140
	Deleting Users	142
	User Groups	143
	Creating a User Group	144
	Deleting a User or a Form from a User Group	145
	User Groups and the Synchronization of Form Designs ...	146
	Deleting Form Designs and Data	146
	Limiting the Number of Records on the Handheld	148
	Switching Off Synchronizing Forms	150
Chapter 8	Advanced Form Properties	153
	Controlling Navigation through a Form	156
	No End Button	156
	No Back Button	157
	No Record View Button	158

Hide Section Skip Button	159
Disable AutoNavigation	159
Default to Field View and Default to Record View	160
Controlling Data Entry	160
Field Level Validation	160
Record Level Undo	161
Controlling the Appearance of Subforms	162
Display as Subform	162
Hide Form in Form List	163
Don't Sort on Subform Screen	163
Automating the Creation of a New Record	164
Auto-Repeat This Form	164
Auto-Execute	164
Controlling Deletion on the Handheld	165
Disable Form Deletion	166
Hide Delete Button	166
Advanced Synchronization Rules	167
Allow Inserts on Handheld to Update Existing Records	167
Storage Limit	168
Archive Date	169
Additional Download Criteria	169

Part III Managing Data and Form Designs 175

Chapter 9	Managing Data on the PC	177
	Where Is the Data from the Handheld Stored?	177
	Viewing and Editing Data in the Database	178
	Datasheet View	178
	Editing a Record	180
	Adding a New Record	180
	Deleting a Record	182
	Alternative Methods of Viewing Data	183
	Creating a Simple Work Order	184
	Creating a Report in Microsoft Access	187
	An Alternative Method for Creating an Access Report	192
	Adding a Filter to a Report	192
	Creating a Report in Microsoft Excel	193
	Exporting Data to ASCII (CSV) Format	193
	Creating a Report in Microsoft Word	194
	Importing Data from an ASCII (CSV) File	196
	Format of a CSV File	196
	Importing a CSV File into a Database Table	197
	Creating a CSV File from Microsoft Excel	198
	Protecting Your Work	199

	Backing Up the Pendragon Forms Database	199
	Backing Up Form Designs	199
	Backing Up Data within a Form	200
Chapter 10	Managing Form Designs	201
	Where Are Form Designs Stored?	201
	Synchronizing to a Specific Form Design	202
	Exporting Form Designs and Lookup Lists	202
	Exporting a Form Design	203
	Exporting a Lookup List	204
	Using Import Features	205
	Importing from a Previous Pendragon Forms Database ...	205
	Importing Data from a Different Form Design	207
	Importing Form Designs and Lookup Lists	208
	Importing a Form Design from a .PFF File	209
	Importing Lookup Lists	209

Part IV Creating Specialized Forms

Chapter 11	Using Scripts	215
	What Is a Script?	215
	Using Event Procedure Names	219
	Calculate Event	220
	Click Event	220
	Enter Event	221
	Exit Event	221
	Initialize Event	221
	Open Event	221
	Select Event	221
	Validate Event	222
	Using Variables	222
	Using Operators	223
	Unary Operators	223
	Binary Operators	224
	Using Statements	225
	Using Assignment Statements to Place a Value in a Field	225
	Testing the User's Response with a Conditional Statement	226
	Using Action Statements in a Script	228
	Using Action Statements with the SPT 1500	236
	Creating Calculation Scripts	236
	Working with Several Fields in a Calculation	236
	Dividing	237
	Rounding	239
	Working with Currency Fields	240
	Creating a Counter	241

	Using Scripts with Dates	241
	Calculating Time Elapsed	242
	Calculating Ages	244
	Adding a Constant to a Date	245
	Branching Scripts	246
	Branching with a Jump Popup Field	247
	Branching from a Popup List or Lookup List	248
	Branching from a Yes/No Checkbox Field	250
	Scripting Button Fields	250
	Creating a Button to Lookup from the Address Book	251
	Creating a Calculate Button	252
	Creating a Button to Add Records of a Form	253
	Creating a Button to Print a Record	254
	Creating a Button to E-Mail a Record	256
	Creating a Button to Send a Record to a Website	257
	Using a Lookup . . . Within Script	257
	Using a Validate Script	260
	Repeating Forms	261
	Printing Scripts	264
Chapter 12	Using Bar Codes	265
	Creating a Form to Accept Bar Code Data	267
	Using an SPT 1500 or SPT 1700	267
	Using a Bar Code Scanner Attached to the Serial Port	268
	Controlling Which Bar Codes Can Be Scanned	270
	Using Scripts to Control Bar Code Input on the SPT 1500 and SPT 1700	270
	Customizing Bar Code Input	274
	Using Bar Code Fields for Inventory Forms	276
	Retrieving a Record by Scanning a Bar Code	277
	Making a Bar Code Field Unique	278
	Library Check-In/Check-Out	279
Part V	Using Pendragon Forms with Existing Databases	
Chapter 13	Linking to an External Access Database	285
	Preparing Your Database Table for Use with Pendragon Forms	286
	Creating a Form from an Existing Database Table	290
	Editing a Form Design for Use on the Handheld	292
	Setting Advanced Field Properties	295
	Controlling Which Records Go to the Handheld	298
	Setting Handheld Access Rights	299
	Setting Additional Download Criteria	300
	Linking the Form to the External Database Table	301

	Sending a Form to the Handheld	307
	Viewing Data on the PC	308
	Linking a Parent Form and Subform to External Database	
	Tables	308
	Order of Synchronization	310
	Linking to an External Database Query	313
Chapter 14	Linking to an ODBC Database	317
	Understanding ODBC	317
	Linking to an ODBC Database	318
	Creating a Linked Table in Access	320
	Mapping Directly to an ODBC Table	325
	Editing the Form for Use on the Handheld	327
	Controlling Which Records Are Sent to the Handheld	327
	Linking to the ODBC Database	327
	Security Concerns When Using ODBC Databases	329

Part VI Working in a Multi-User Environment

Chapter 15	Planning a Multi-User Installation	333
	Configuring Pendragon Forms for Multiple Users	333
	Deciding Whether Records Are Shared	335
	Enabling Remote Access	336
	Choosing the Right Synchronization Option	337
	Synchronizing to One PC	337
	Synchronizing Over a LAN Using Pendragon Forms	
	Network Setup	337
	Synchronizing Over a LAN Using 3Com's Network HotSync	339
	Using the WaveSync Synchronization Server from	
	WaveWare Communications	341
	Security Issues to Consider	343
	Taking Precautionary Measures	344
	Security Limitations	345
Chapter 16	Working with the Palm VII	347
	How the Palm VII Works	347
	Choosing Pendragon Forms or PQAs	348
	Creating a Transmit Button	349
	Designing a Palm VII-Compatible Web Site	354
	 Appendix A: Troubleshooting Tips	 355
	 Appendix B: Scripting Syntax	 375
	 Appendix C: What's on the CD-ROM?	 379

Glossary 393

Index..... 399

End-User License Agreement 427

CD-ROM Installation Instructions 432

Part I

Creating Customized Forms for the Palm Organizer

IN THIS PART

CHAPTER 1

What Is Pendragon Forms?

CHAPTER 2

Creating a Form

CHAPTER 3

Entering Data on the Palm Organizer

Chapter 1

What Is Pendragon Forms?

IN THIS CHAPTER

- ◆ Using the Palm Computing platform
- ◆ Learning how Pendragon Forms works
- ◆ Deciding if Pendragon Forms is the right development tool
- ◆ Using this book and CD-ROM
- ◆ Installing Pendragon Forms

PENDRAGON FORMS IS A software package for building database applications for Palm Computing organizers, the family of personal digital assistants that includes the best-selling 3Com PalmPilot and Palm III devices.

This chapter looks at why the Palm Computing platform is a good choice for deploying an application to a mobile workforce, and why Pendragon Forms is a good choice for rapid development of your own custom Palm application.

The Palm Computing Platform

The Palm Computing Platform is a collective name for the Palm operating system (OS) and the hardware devices on which it runs.

The Palm Computing platform is the most successful handheld platform ever. There are several reasons for this success: size and battery life, the Graffiti handwriting recognition system, and HotSync technology.

Size and Battery Life

The 3Com Palm organizer is a pocket-sized device with excellent battery life. It is not unusual for Palm devices that run on AAA batteries to last a month compared with a battery life of less than a week on competing handheld platforms.

Interestingly, the electronics technology used in the first few generations of the platform was not revolutionary. The central processing unit (CPU) was a low-power, integrated variant of the Motorola 68000, the same chip that powered the original Macintosh more than ten years earlier. In fact, it was the slow speed of the CPU that

made the long battery life possible, because the faster a CPU runs, the more power is required.

What makes Palm Computing devices revolutionary is the software. The Palm OS software that runs on the handheld is designed to be “lightweight” so that applications are very responsive, even without a high-performance CPU. Even today, although the newer Palm devices such as the Palm V and Palm VII, run on a different CPU (the Motorola DragonBall EZ), the clock speed is still 16MHz – the same as the original Palm Computing organizer. Competing handheld platforms that run the Windows CE operating system run on CPUs with clock speeds of 75-133MHz, but to the handheld user, these devices do not appear to be faster than a Palm device because of the responsiveness of the Palm OS.

Graffiti Handwriting Recognition

The Palm OS features a handwriting recognition system called *Graffiti*. Graffiti is an easy-to-learn writing system similar to normal block lettering. Its success is due to the fact that it places the burden of writing on the user instead of on the handheld, and also requires less computing power than devices that interpret the user's natural writing style.

HotSync Technology

The other key component of the Palm Computing platform is the HotSync technology. The HotSync mechanism allows a Palm device to synchronize with a desktop PC at the touch of a button. Palm Computing handhelds include a cradle that plugs into a PC. The cradle has a single button for initiating the HotSync data transfer.

The HotSync Manager software that runs on the PC is an extensible system, allowing third-party applications to “plug in” to the system. Once plugged in, applications participate in the HotSync data transfer when the HotSync button is pressed.

As a personal organizer, a Palm Computing device combines small size, long battery life, ease of use, and the ability to synchronize with a desktop PC. This has proven to be a winning combination; over four million Palm Computing devices have been sold. The popularity and low cost of the device has also made it an attractive platform for third-party developers, who have created a wide range of programs that expand the original functionality of the Palm organizer.

Types of Palm Devices

Several Palm Computing devices are on the market today – most are manufactured by the Palm Computing division of 3Com Corporation. 3Com has also licensed the right to develop Palm Computing devices to several other key partners, who have created their own handheld devices that run on the Palm OS. These devices include the IBM Workpad, the Symbol Technologies SPT 1500 bar code scanning solution, the Qualcomm pdQ Smart Phone, and the HandSpring Visor.

Table 1-1 shows a comparison of the Palm Computing devices.

TABLE 1-1 PALM OS DEVICES

Product Name	Description	Standard Memory
3Com Palm III	The current base model has an infrared port, a serial port, and 160x160 pixel display.	2MB
3Com Palm IIIx	This model has the same dimensions as a Palm III, but has an internal slot for memory expansion and a high-contrast display.	4MB
3Com Palm IIIe	The entry-level version of the Palm IIIx ships with 2MB of memory.	2MB
3Com Palm V	The smallest Palm OS device to date, this model incorporates a high-contrast LCD display in an anodized aluminum case. The Palm V has a built-in rechargeable battery.	2MB
3Com Palm VII	Slightly larger than the Palm III, this model incorporates wireless connectivity for executing Web transactions and lookups.	2MB
IBM Workpad	IBM offers its own branded versions of the Palm V and the Palm IIIx.	2MB /4MB
Qualcomm pdQ Smart Phone	A CDMA cellular phone with a built-in Palm III, enabling wireless access for Palm applications.	2MB
Symbol Technologies SPT-1500	Based on the Palm III, this model incorporates a laser barcode scanner into the top of the handheld.	2MB
Symbol Technologies SPT-1700 and SPT1740	The SPT 1700 is a rugged version of the SPT 1500. The SPT 1740 has wireless local area networking.	2MB (expandable to 8MB)
Handspring Visor and Visor Deluxe	Palm OS computer with Springboard expansion slot for adding hardware and software modules.	2MB (Visor); 8MB (Visor Deluxe)

What Is Pendragon Forms?

Pendragon Forms is a database application that enables you to create your own custom forms for your Palm organizer.

The capability of customizing a form – a template that collects fields of information – extends the functionality of your Palm organizer. In addition to using the built-in Palm applications such as the Address Book, Date Book, and Memo Pad, you can customize forms for surveys, inspections, patient tracking, work orders – the list is endless.

Originally launched by Pendragon Software in November 1996, Pendragon Forms was one of the first commercial products for the Palm Computing platform to utilize a conduit and the HotSync technology. The conduit gives the handheld the capability to send data to the PC, and once the data has been uploaded to the PC, it is then possible to generate reports and analyze data collected by mobile workers.

How Does Pendragon Forms Work?

Pendragon Forms consists of three main components:

- ◆ **The Pendragon Forms Manager.** This is a database that runs on your PC. The Pendragon Forms Manager is used to create forms, manage which forms are sent to the handheld, and view data that is sent from the handheld to the PC. All forms that you create are stored in this database, which is stored in Microsoft Access format. For each form that you send to the Palm organizer, Pendragon Forms creates a separate database table within the Pendragon Forms Manager for storing data associated with the form.
- ◆ **The Pendragon Forms program for Palm OS.** This is an application that you install on your Palm organizer. The Pendragon Forms program interprets form designs that are sent from the PC, and displays the forms so that you can fill in the fields.
- ◆ **The Pendragon Forms conduit.** This is a plug-in to the HotSync Manager that enables forms, and data associated with those forms, to be synchronized during the HotSync data transfer. During a HotSync data transfer, the HotSync Manager synchronizes the built-in applications and then gives control to the Pendragon Forms conduit to synchronize forms and data.

Creating a working form starts in the Form Designer in the Pendragon Forms Manager. The Form Designer presents you with simple screens for entering the fields on your form and setting their attributes.

Before a form design can be sent to the Palm device, you must freeze the form. Freezing a form creates the associated database table in the Pendragon Forms

Manager database for storing the records that are created on the handheld. The form can then be selected for distribution to the handheld.

During a HotSync data transfer, the Pendragon Forms conduit sends the form design to the Palm organizer. After the HotSync data transfer is complete, the Pendragon Forms program on the handheld displays a list of forms that were sent to the Palm device. Filling in a form creates a record on the handheld. You can create new records or view and modify existing records for each form on the handheld. Whenever you perform a HotSync data transfer, new and changed records are sent to the PC.



Pendragon Forms version 3 supports bi-directional synchronization, meaning that if records are entered on the PC they are sent to the handheld during the HotSync data transfer. Many applications use this capability to send reference data or work orders from the PC to the Palm device.

Figure 1-1 shows the relationship between the Pendragon Forms Manager on the PC, the Pendragon Forms conduit, and the Pendragon Forms program on the Palm device.

Deciding if Pendragon Forms Is the Right Development Tool

Users of Pendragon Forms don't need programming skills to create their own custom forms. This makes the product ideal for those who want to create and deploy a Palm application with a minimal development cycle. A form with ten fields can take as little as ten minutes to create with Pendragon Forms.

However, in making a product that is as easy to use as Pendragon Forms, some flexibility and performance had to be sacrificed. It is not possible to access all the features of the Palm organizer, or to build the fastest applications with Pendragon Forms.

The best way to create a software solution for any given application is to write a custom program in a low-level language such as C or C++. These languages produce the smallest and fastest programs. Unfortunately, C and C++ are hard to learn and use. Learning to develop for the Palm OS can take a couple of months even for skilled developers. Even after the low-level programming tools are mastered, it can still take many hours just to produce a single screen to display on the handheld, and writing conduit software for synchronizing an application with a desktop or server can be more complex than building the program for the handheld. A typical handheld application consisting of a Palm OS program and a conduit takes months of development time by skilled developers, and is therefore very expensive.

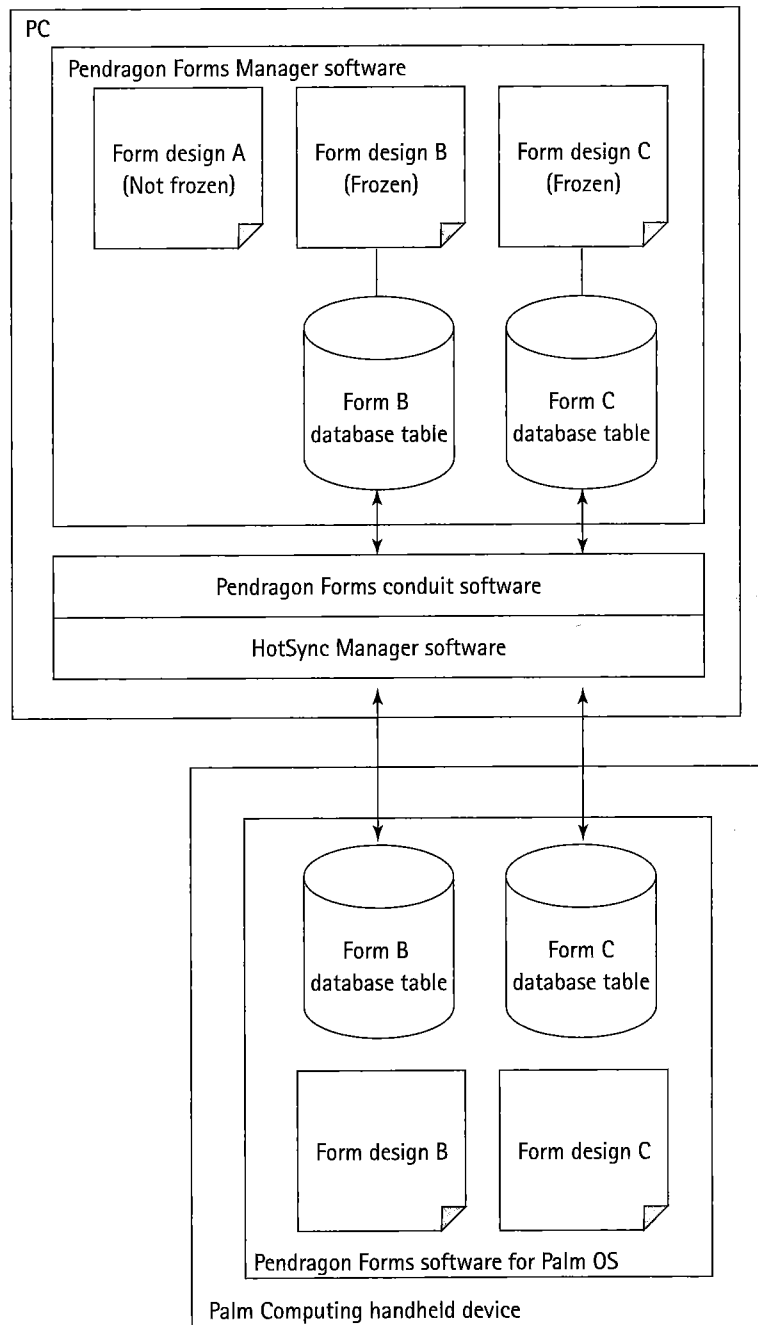


Figure 1-1: Relationship between the Pendragon Forms components

Pendragon Forms is designed to provide the answer to a wide range of development needs with little or no programming, and may be the only option for non-programmers on a tight budget. For example, a field service work order application that might take three months of development in C++ can be built in just a couple of hours with Pendragon Forms.

So when should you use the low-level programming tools instead of Pendragon Forms? The easiest way to answer this question is to try creating a form with Pendragon Forms to see if it meets your data collection needs. If after a few hours of experimentation you are unable to get the functionality you need or the performance you require, then you probably need to look at other development tools. Some factors affecting this decision are listed as follows:

- ◆ Large databases that maintain thousands of records on the handheld may not be appropriate for Pendragon Forms. Managing this volume of data may require special programming techniques with low-level tools and may require a custom conduit to reduce synchronization time.
- ◆ Pendragon Forms automatically handles the layout of the handheld screens for you. Custom screen layouts require the use of other development tools.
- ◆ Pendragon Forms can synchronize with various databases and enables you to extend its capabilities by writing your own programs on the PC with popular tools such as Visual Basic. However, if you want to synchronize with non-database formats such as HTML, you will likely need to work with other tools.

Note that even when the final application must be written with another development tool, it may still be helpful to build a prototype application with Pendragon Forms. Prototypes built with Pendragon Forms are excellent tools for testing the feasibility and usability of handheld applications.

How to Use This Book and CD-ROM

The chapters in this book progress from the basics that you need to know to create a Pendragon form, to specialty topics such as using scripts to perform calculations. Later chapters in this book discuss advanced topics, such as linking to an external Microsoft Access database or configuring Pendragon Forms for a multiuser installation.

The CD-ROM in the back of this book contains a 30-day evaluation version of Pendragon Forms that you can install. Once installed, the product contains sample forms that are referenced throughout the chapters of this book. You can distribute any of the sample forms to your Palm organizer. If you want to continue using Pendragon Forms after the 30-day trial period, you can refer to the CD-ROM for instructions on upgrading to the full version of Pendragon Forms.

The CD-ROM also contains evaluation software for several products that work in conjunction with Pendragon Forms.



If you currently use Pendragon Forms version 3.0, you do not need to install Pendragon Forms from this CD-ROM. To use the sample forms, open the Pendragon Forms Manager and click Import Forms/Data/Lookups. Open the Forms3.mdb file in the Database folder on the CD-ROM, and then select a form to import.

Getting Started: Installing Pendragon Forms

To install the evaluation version of Pendragon Forms that comes with this book, do the following (see the sidebar for your system requirements):

1. Close any applications not in use, especially the HotSync Manager and the Microsoft Office toolbar. (To close the HotSync Manager, right-click the icon with the red and blue arrows in the system tray at the right side of your Windows Taskbar, and choose Exit.)
2. Insert the CD-ROM into the CD-ROM drive. If the CD-ROM does not automatically start, click the Windows Start button, and then choose Run. Type `D:\SETUP`, where *D* is the drive letter referring to your CD-ROM drive. Press Enter.
3. Click Install Pendragon Forms.
4. A Pendragon Forms Installation dialog box will prompt you to choose which version of Microsoft Access you have, if any. If you agree with the default option, click OK. Alternatively, click the correct option, and then click OK.
5. Follow the on-screen prompts to complete the installation.
6. After installing the software, you may need to restart your PC.
7. To open the Pendragon Forms Manager on the PC, click Start → Programs → Pendragon Forms 3.0 → Pendragon Forms Manager. You will be prompted to enter your Palm user name. (If you do not know your Palm user name, tap the HotSync icon on the handheld, and the user name will appear in the upper right corner of the handheld screen.)
8. To install Pendragon Forms on the Palm device, click Start → Programs → Pendragon Forms 3.0 → Install Forms 3.0 on Handheld.
9. The Palm Install Tool will run. Verify that your handheld user name is selected, and click OK. Verify that the Forms3.prc program is selected. The Forms3.prc program is typically in the `C:\Program Files\Forms3` folder.

10. Click Done on the Palm Install Tool, and then perform a HotSync data transfer. After synchronizing, tap Applications on the handheld to verify that the Forms icon is present.

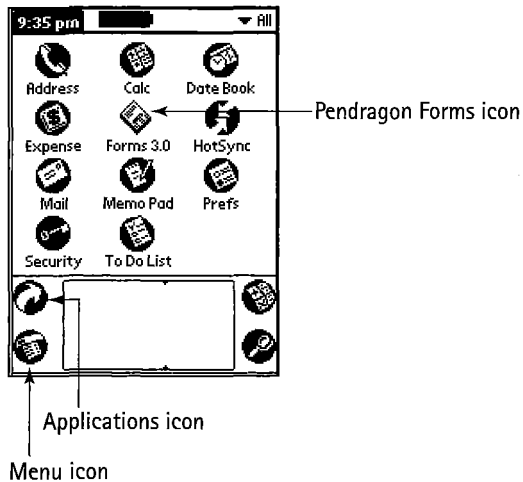


Figure 1-2: Pendragon Forms on the Palm organizer



Refer to Appendix A, "Troubleshooting Tips," if you have installation problems.

System Requirements for Pendragon Forms

On the PC, Pendragon Forms requires Windows 9x or Windows NT4.0 (with Service Pack 3 or higher). A minimum of 32MB RAM and 25MB free hard disk space are required for installation. Microsoft Access 97 or later is not required, but is recommended. If you do not have Microsoft Access, note that a run-time version of Access 97 is installed when Pendragon Forms installs.

On the handheld, Pendragon Forms requires Palm OS 3.0 or higher, 2MB RAM (Total Memory), HotSync Manager 3.0 or higher. Pendragon Forms works with all handheld devices referenced in Table 1-1. The Forms3.prc program on the Palm takes up 165KB of memory on the handheld, before any form designs or data is added.

Summary

This chapter outlined the Palm Computing platform and how Pendragon Forms works to enable you to create custom applications for Palm organizers. The chapter described the installation of the evaluation version of Pendragon Forms that comes on the CD-ROM with this book, and explained how the sample forms on the CD-ROM can be used with this book.

Chapter 2

Creating a Form

IN THIS CHAPTER

- ◆ Creating a Pendragon form
- ◆ Sending a form to the Palm Organizer
- ◆ Viewing data on the PC
- ◆ Changing form designs

THIS CHAPTER WILL WALK you through the creation of your first form design. Many of the finer points of form design are covered later, but this chapter will help you get a feel for the big picture.

In Pendragon Forms, all forms are designed in the Pendragon Forms Manager on the PC before being sent to the handheld. There are three steps in the design process:

1. Designing (creating) a form
2. Freezing the form design
3. Distributing the form to the handheld

Once you have sent a form to the handheld, you can enter data on the handheld and upload data to the PC.

Designing a Form

In the Pendragon Forms Manager on the PC, you use the Form Designer window to create a form. Every form that you create has a name, and for each item of data that you want to collect on the form, you need to create a separate field. A field has a name and a field type. The field name prompts the handheld user for the input that is needed in a field. (For example, a field name called Customer Name is asking the user to enter the name of a customer in that field.) The field type determines the type of data that you can enter in that particular field.

To access the Form Designer window, click the New button in the Pendragon Forms Manager, as shown in Figure 2-1.

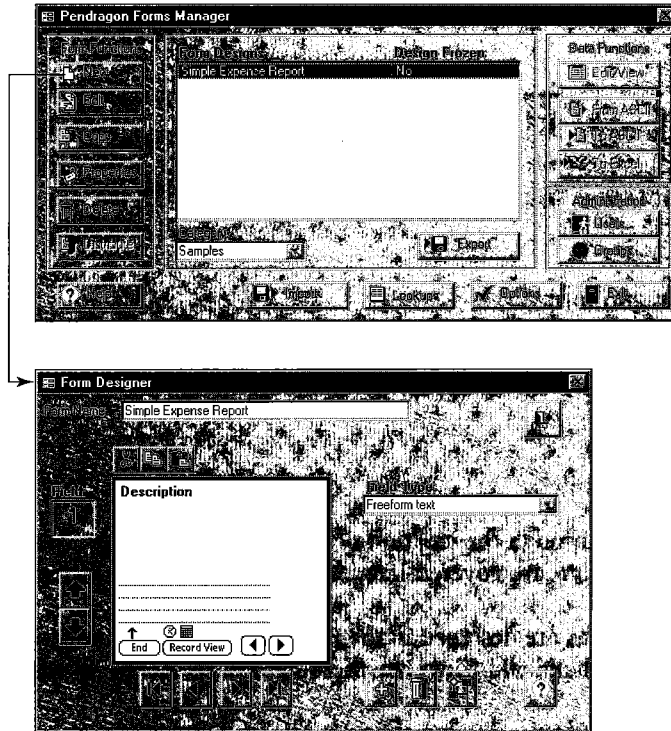


Figure 2-1: Clicking the New button brings up the Form Designer window.

The Form Designer window includes a Form Name field for you to type a name for the form. For each field that you want to add to your form:

1. Type a name for the field in the Field area of the Form Designer window. The field name tells the handheld user what type of information is expected in the field.
2. Select a field type for the field. The field type determines what type of data the handheld user can enter. You can choose from 21 field types.



See Chapter 5 for a complete list of field types.

3. Depending on the field type that you select, you need to enter additional information. For example, a Popup List is a field type that displays a list of items on the handheld. When you select Popup List as the field type, a

Popup Options box appears for you to enter the items that you want the handheld user to see in the list.

4. Click the + button to add another field on the form. A form can have up to 250 fields.



If you are viewing the last field of a form, you can also click the right arrow button in the Form Designer window to add a new field to the end of the form.



If you have a form with 250 fields, you will not be able to store as many records on the handheld as, say, with a form that has 10 fields. Moving from one field to the next will also be slower, but not enough to make the form unusable.

5. Click the Close button to close the Forms Designer and save your form design. (You will not be prompted to provide a filename. All form designs are stored internally within one database file, the forms3.mdb file in the C:\Program Files\Forms3 folder. If you are using Microsoft Access 2000, the filename is forms32k.mdb) Your form design is not saved until you click this button.



The Simple Expense Report form on the CD-ROM is used in the following example to illustrate the process of designing a form. If you open the Pendragon Forms Manager, click the arrow next to the Category field, and select the Samples category to view the sample forms mentioned in this book. In this example, if you click the Simple Expense Report form, and then click the Edit button, you can take a look at this sample form.

To illustrate the rationale behind selecting different field types for different fields, let's look at the four fields that make up a Simple Expense Report form. Each field name tells the handheld user what type of data is expected to be entered in the field, and a field type is selected to make data entry as quick and as accurate as possible. For example, if you want the handheld user to enter a date, use a date field type, which forces the handheld user to enter a date – the field will accept no other type of input. This speeds up data entry because the handheld user uses fewer Graffiti strokes and more stylus-tapping selections.

The first field on the expense report is a Description field, shown in the bottom part of Figure 2-1. A Freeform Text field is used as the field type, because you need to be able to enter anything. The description might be “Business Trip to See XYZ Company,” or “Annual Company Conference.” A Text field, as it is usually referred to, allows you to enter up to 255 characters on the handheld.



See Chapter 6, “Advanced Field Properties,” if you need to enable the handheld user to enter more than 255 characters in a Text field.

The next item of information that you may need to record on the expense report is the date on which the expense was incurred. Figure 2-2 illustrates a Date of Expense field. To ensure that a date can be entered in this field only, select a Date Only field type.

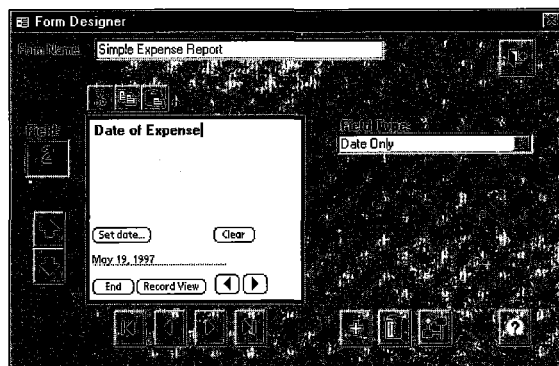


Figure 2-2: Selecting a Date Only field

Because travel expenses generally fall into fixed categories, a Popup List is used in the next field, shown in Figure 2-3. On the handheld, a Popup List lets the user select an item from the list with one tap of the stylus, thus avoiding the need to enter whole words. In this example, the Popup List has options Air, Car, Meal, Hotel, Taxi, Tips, Other.

A Popup List can store up to 512 characters for the entire list. Each item in the list must appear on a separate line, in order to make it possible to select an individual item with one tap of the stylus on the handheld. Pressing Enter to add a new line counts as 2 characters. In Figure 2-3, the pop-up list containing the options Air, Car, Meal, Hotel, Taxi, Tips, Other takes up a total of 40 characters – 28 characters for all the letters in the list, and pressing Enter six times adds 12 characters.



In this simple example, the purpose of offering a choice such as “Other” in a Popup List is to record all miscellaneous expenses in one category. In a more sophisticated form design, you may want to add a Text field to allow the user to record what type of expense is meant when ‘Other’ is selected. See Chapter 11, “Using Scripts,” for an example of creating a form that branches to an additional field, depending on the selection made in a Popup List.

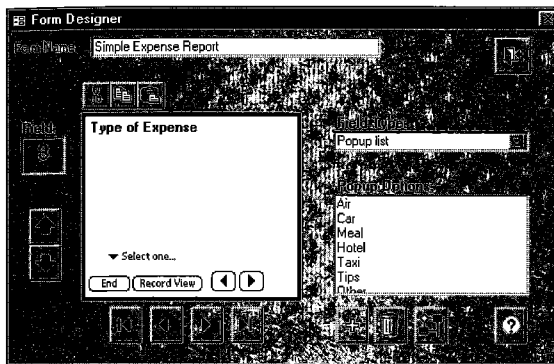


Figure 2-3: A Popup List field

Create an Amount field to record the dollar amount of the expense. Figure 2-4 shows that a Currency field is used as the field type, to ensure that numbers are entered as dollars and cents.

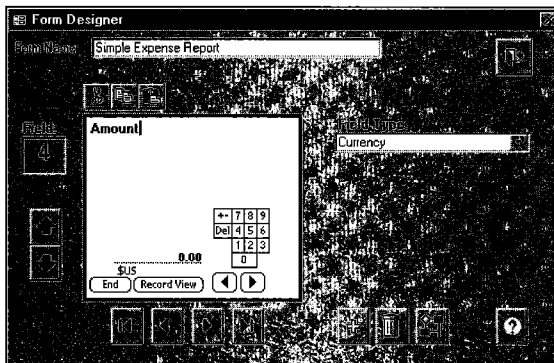


Figure 2-4: A Currency field

What's Important about the First Field on a Form?

By default, the first field on a form is used as a Display Key field. This means that when you are reviewing records on the handheld, you will see the first field of each record displayed. In order to distinguish between different records, you should make the first field on a form a field that is generally unique across records.

As the following figure illustrates, a Popup List and a Yes/No check box are not recommended as the Display Key, because many records can have the same value.

The screenshot shows a form titled "Records" with a dropdown menu set to "Default". Below the title is the form name "Work1". The list of records has the following first field values: N, Y, N, Y, Y, Y. At the bottom of the form, there is a "All Fields" dropdown menu, a search icon, a close icon (X), and a "Done" button. The "TimeStamp" field is visible at the bottom left.

A Yes/No field or a Popup List is not a good choice for the first field on a form.

Text fields, Numeric fields, and Date fields are more likely to be unique across records. The following figure shows a better choice for the Display Key, which makes it easier to tell records apart.

The screenshot shows a form titled "Records" with a dropdown menu set to "Default". Below the title is the form name "Work2". The list of records has the following first field values: Caldwell, Logan, Ramsaran, Sampson, Jones, Smith. At the bottom of the form, there is a "All Fields" dropdown menu, a search icon, a close icon (X), and a "Done" button. The "TimeStamp" field is visible at the bottom left.

A Text field is a good choice as the first field on a form, because records are easy to distinguish.














If you cannot make the first field on your form a unique field, you can choose to change which field is the Display Key, by setting the Display Key property of a field. See Chapter 6, "Advanced Field Properties."

Editing a Form Design

You can edit form designs in the Form Designer window. Note that changes are not saved until you close the Form Designer window.

To edit a form design, click the name of the form, and then click the Edit button. The editing tools available in the Form Designer window are shown in Table 2-1.

TABLE 2-1 EDITING BUTTONS IN THE FORM DESIGNER WINDOW

Button	Description
	Add a new field. The new field is added after the currently displayed field.
	Delete the currently displayed field.
	Move a field earlier in the sequence on the form. For example, move field 10 into position 9.
	Move a field later in the sequence on the form. For example, move field 16 into position 17.
	Display the First field on the form.
	Display the Previous field on the form.
	Display the Next field on the form. If you are on the last field of the form, this button will create a new field at the end of the form.
	Display the Last field on the form.
	Cut the current field out of the form and place it on the Pendragon Forms clipboard (not the Windows clipboard).
	Copy the current field to a clipboard.
	Paste the contents of the clipboard into the field after the currently displayed field.
	Save the form design and close the Form Designer window.
	The Advanced Field Properties button is used to set advanced field properties and create scripts.

Freezing a Form Design

Once you have created a form design, you need to freeze the form design before you can send the form to the handheld. The Form Properties window is where you freeze the form design, and where you specify how long you want records to remain on the handheld.



When you freeze a form, you make certain aspects of it read-only. A database table has to be created to store the records associated with the form, and three aspects of the form become fixed ("read-only") when you freeze a form—the number of database columns, the names of the database columns, and the data types used in each column (such as Text or Numeric). Other aspects of a form can be changed—the form name and the names of the fields, for example.



For information on the types of changes you can make to a form after it has been frozen, refer to the topic "Making Changes to a Frozen Form" in this chapter.

Freezing a form design creates a database table in the Pendragon Forms Manager database on your PC for storing records associated with the form. Each column in the database table corresponds to a field on your form. Column names are derived from the first 60 characters of your field names, minus spaces and punctuation.

The Data Persistence section of the Form Properties window is used to determine how long records will remain on the handheld. Because the memory on the handheld is much smaller than the storage space on a PC, setting an appropriate Data Persistence option is an important part of managing data on the handheld.



See Chapter 4, "Planning a Form Design," for more information on this topic.

Once a form is frozen, you cannot add or delete fields, but you can change Data Persistence options. If you freeze a form and then realize that you need to add or delete fields, you can make a copy of the form. A copy is not frozen and can be modified in any way.



Copying a form is similar to doing a Save As, with two exceptions — the copy contains the form design only, not any data that has been collected in the original form. Second, the copy is not frozen.

To access the Form Properties window, click the Properties button in the Pendragon Forms Manager, shown in Figure 2-5.

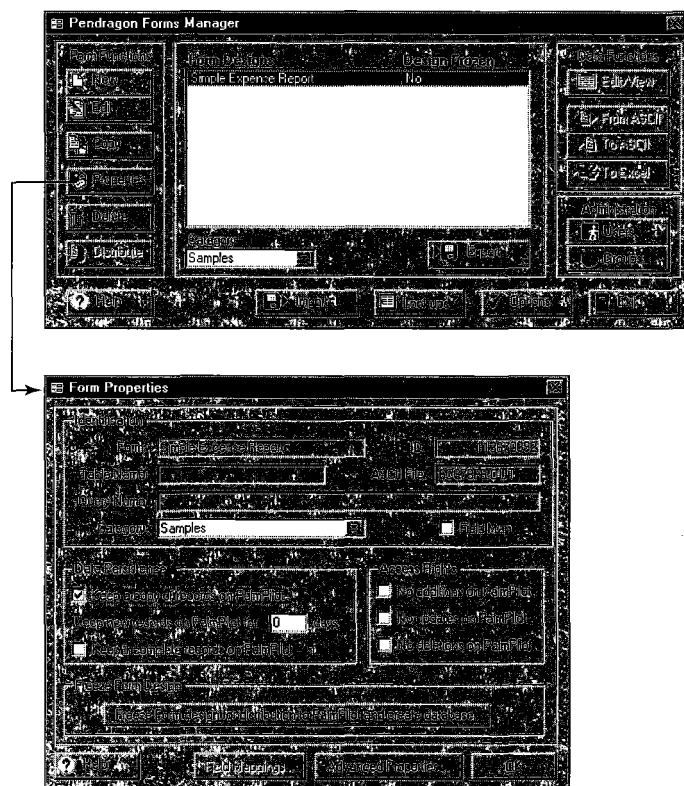


Figure 2-5: Clicking the Properties button brings up the Form Properties window.

The Form Properties window includes a Data Persistence section that is used to determine how long records will remain on the handheld. Select one of the following options:

- ◆ **Default option** – No Data Persistence check boxes checked. After a HotSync data transfer, records are removed from the handheld. This option is useful if you are entering large amounts of data on the handheld that you do not need once the records have been sent to the PC.

- ◆ **Keep a Copy of Records on Handheld.** If you check this box, then records will be uploaded to the PC during synchronization, but a copy of the records will remain on the handheld after the HotSync process. This option is recommended if you want to back up information to the PC but you also need to keep records with you in the mobile handheld environment. This option is also useful if you are testing form designs and you need to compare data on the handheld with data on the PC.
- ◆ **Keep New Records on Handheld for X Days.** If you enter a number of days from 0 to 999, then after a HotSync data transfer, records will continue to be stored on the handheld for the specified number of days. This is useful if you need to refer to records on the handheld for a limited time, such as a week or a month, and beyond that time you no longer need the outdated records on the handheld.
- ◆ **Keep Incomplete Records on Handheld.** This option requires a Completion Checkbox field on your form in order to work. (If you know you will need this option, add a Completion Checkbox field to your form when you design the form.) If you check the Keep Incomplete Records on Handheld option, then only those records with the Completion Checkbox checked will be removed from the handheld. This option is useful if you cannot predict the exact number of days that you will need records on the handheld and so you want to control when individual records are removed.



You can change Data Persistence options at any time. For example, if you want to change from keeping all records on the handheld to keeping records for 5 days only, uncheck the Keep a Copy of Records on Handheld check box, and instead enter the number 5 in the option Keep New Records on Handheld for X Days. If the form is already on the Palm organizer, and you change a Data Persistence option, you will need to redistribute the form to the handheld for the change to take effect. To redistribute a form design, close the Properties window, click the name of the form in the Forms Manager, and then click the Distribute button.

Click the Freeze Form Design button to freeze the form design. You will be prompted to confirm this action. Freezing a form design creates a database table in the Pendragon Forms Manager database for storing the records that are created when you fill out the form on the handheld. The database table includes one database column for each field on your form, plus four columns that are used internally by Pendragon Forms. You can freeze a form only once. After the form design has been frozen, you can close the Form Properties window.

After you freeze a form design, you will notice that fields in the Identification section of the Form Properties window are automatically filled in (see Figure 2-6). The Table Name corresponds to the name of the database table that has been created for storing records associated with the form.

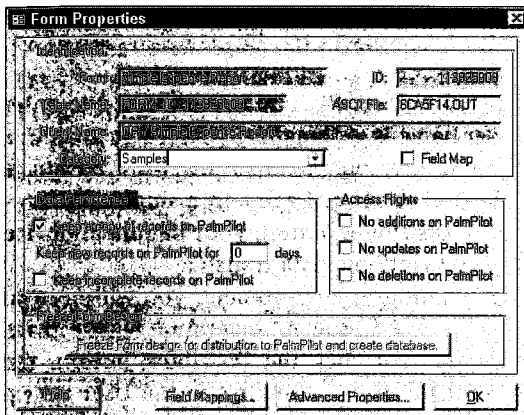


Figure 2-6: Fields in the Identification section of the Form Properties window are automatically filled in after you freeze a form design.

Organizing Form Designs into Categories

The Form Properties window contains a Category field that you can use to organize your form designs. A category is like a folder used for storing form designs. For example, if you are designing several forms for one data collection project, you may want to place all the forms related to that project in a category of their own.

When you create a new form, the default category is Unfiled. If you type a category name in the Properties window, a new category will be created, and the form design will be assigned to that category. Once a category has been created, other forms can be placed into that category by clicking the name of the form, clicking the Properties button, and selecting the category to which you want to assign the form.

You can choose to display the form designs in a given category by clicking the Category field in the Pendragon Forms Manager window. The All category shows all form designs that you have created. The Samples category is a special category used for sample forms that are mentioned in this book. Because you are not likely to need the sample forms once you are familiar with using Pendragon Forms, the forms in the Samples category do not appear when you select the All category.

Sending a Form to the Handheld

Once a form design has been frozen, you can send the form to the handheld. In the Pendragon Forms Manager, you use the Distribute button to mark a form for distribution to the handheld during the next HotSync data transfer.

For a form to be sent to the handheld, the form must be assigned to a group in which the handheld is a member. In the single-user case, this is taken care of automatically.

When you first install Pendragon Forms on your PC, you are prompted to enter a handheld user name. This handheld user name is automatically added as an active user in the Default User Group. When you distribute a form, the form design is marked for distribution to all users in the Default User Group. In a single-user scenario, this means that when you click the Distribute button, your handheld device will receive the form design during the next HotSync data transfer.

If you are using Pendragon Forms in a multiuser environment, you will need to add users to the active User List and set up User Groups to determine which forms are sent to which handheld devices. When you click the Distribute button, the selected form will be sent to all the handheld users belonging to the User Group that contains the form.



Chapter 7, "Synchronization Rules," explains how to set up Users and User Groups.

Figure 2-7 shows that you must select a frozen form in order for the Distribute button to be active. Click the name of a form to select the form, and then click the Distribute button. A dialog box informs you that the form has been updated for distribution.

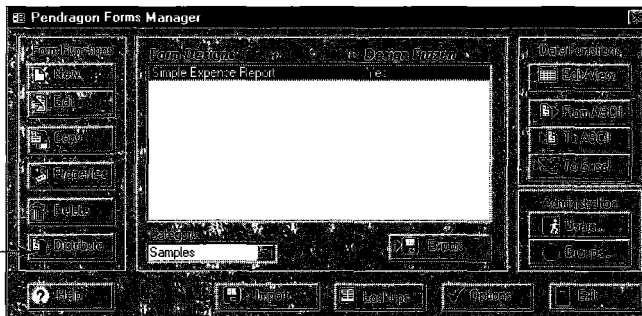


Figure 2-7: The Distribute button in the Pendragon Forms Manager



You can distribute more than one form at a time by repeating the process of selecting a form and then clicking the Distribute button.

Perform a HotSync data transfer. During the synchronization process, you will see the message *Synchronizing Pendragon Forms*. This indicates that the Pendragon Forms conduit is active and is sending form designs to the handheld.

After the HotSync process is complete, tap the Forms icon on the handheld. The form you designed will appear in the list of forms. If you do not see your form on the handheld, refer to Appendix A, "Troubleshooting Tips."

Entering Records on the Handheld

Once you have sent a form to the handheld, you can begin entering records. Tap the Forms icon, and a Forms List screen shows the form designs that have been sent to the handheld.

As shown in Figure 2-8, the Forms List screen contains a New button for creating new records and a Review button for reviewing records that have previously been entered on the handheld or that have been downloaded from the PC.

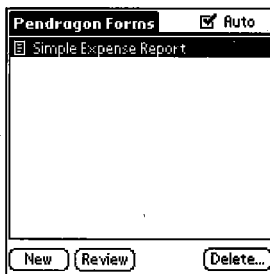


Figure 2-8: The Forms List screen on the handheld

To create a new record on the handheld, tap the name of a form, and then tap the New button to create a new record for the selected form.

By default, the form is displayed in Field View. As shown in Figure 2-9, Field View displays one field on the handheld screen at a time. Enter the required information for the field, and then tap the right arrow button to move to the next field on the form.

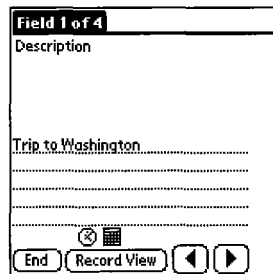


Figure 2-9: Field View on the handheld

When you have stepped through all the fields on the form, you will return to the Forms List screen. You can also choose to exit a record at any time by tapping the End button.



You may exit without completing all fields, unless the form was created with all fields set to Required. See Chapter 6, "Advanced Field Properties," for information on making fields Required.

If you leave the Pendragon Forms application without tapping the End button, for example, by tapping the Applications button on the handheld and then tapping the Address Book application, when you return to the Pendragon Forms application, you will be in the same form and same field that was being displayed when you left the Pendragon Forms application.

Using Field View and Record View

Pendragon Forms can show you two views of a form on the handheld. Field View, seen in Figure 2-9, displays one field at a time. Record View, shown in Figure 2-10, displays several fields at a time in a two-column format – field names in the left-hand column, and responses in the right-hand column. Record View can display up to 11 fields at once.

Trip to Washington	
Description	Trip to Washington
Date of Expense	9/27/99
Type of Expense	Hotel
Amount	\$150.00

End ⏪ ⏩ ⏴ ⏵

Figure 2-10: Record View on the handheld

Field View is useful if you want to step through a form in sequence. Because only one field is displayed at a time on the handheld screen, you can use the field name to display detailed instructions to the handheld user. Field View is also used when writing branching scripts. Because the handheld user sees only one field at a time, branching scripts can be used to select which field is displayed next, depending on the input in the current field.



See Chapter 11, "Using Scripts."

Record View is useful if the information that you are recording is primarily in Yes/No check boxes or Popup Lists. By displaying eleven fields on screen at once, you can easily enter information with a minimum of stylus taps. Record View is also useful if you use the same form repeatedly and you do not need to see the full names of the fields because you are familiar with the form.

You can switch between Field View and Record View at any time.

- ◆ If you are in Field View, tap the Record View button to switch to Record View.
- ◆ If you are in Record View, tap the name of a field in the left-hand column to switch to Field View for that field.

Uploading and Viewing Data on the PC

After you have entered new records on the handheld, you can perform a HotSync data transfer to upload the data to the desktop PC.

Naming Fields for Use in Both Field View and Record View

A field name can contain up to 255 characters. However, if the field name extends past five lines in the Form Designer window, the portion that scrolls out of view will not be visible on the handheld.

As the following figure illustrates, a long field name can be useful in Field View to display instructions to the handheld user. However, the same field name shown in Record View, as in the second figure, can be unclear to the handheld user.

A long field name, as displayed in Field View

The same field name truncates poorly in Record View

The first line of a field name is significant—when you view a form on the handheld in Record View, only characters on the first line (or up to a colon on the first line) of a field name are displayed. By entering an abbreviated field name on the first line of the field name, with additional instructions on the second line, you can create a field whose name is readable in both Field View and Record View. (Note that pressing Enter counts as two characters in the field name.)

The following figure demonstrates a long field name that looks good in Field View and has been formatted so that the same field is readable in Record View also, as seen in the next figure.

Field 2 of 5			
Temperature: Enter the ambient temperature.			
	+-	7	8
Del	4	5	6
	1	2	3
	0	.	
Enter a number			
End	Record View	◀	▶

6/4/99	
Date	6/4/99
Temperature:
Location:
Number of birds:
Comments:
End	
◀◀ ▶▶	

A long field name that has been formatted for use in Record View is still readable in Field View.

In Record View, field names that have been formatted appropriately are readable.

To view the data that has been uploading to the PC:

1. Click Start → Programs → Pendragon Forms → Pendragon Forms Manager.
2. Click the name of a form.
3. Click the Edit/View button (see Figure 2-11).

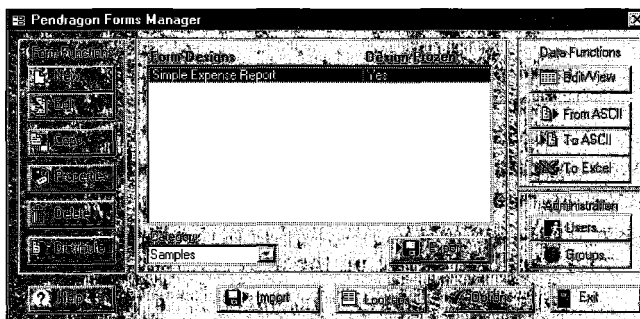


Figure 2-11: The Edit/View button in the Pendragon Forms Manager

As Figure 2-12 illustrates, an Access form is displayed in Datasheet view to allow you to view and modify your data. (The entire Pendragon Forms program on

the PC is a Microsoft Access database.) Datasheet view displays your fields in database columns, and your records in rows.

RecordID	UnitID	UserName	TimeStamp	Description	DateCreated	TimeCreated	Amount
0	0	Sarah Jane	399 6:37:31 PM	Trip to Washing	09/27/1999	Hotel	\$150.00
0	0	Sarah Jane	39 10:36:25 AM	Trip to Washing	09/27/1999	Meal	\$24.36
0	0	Sarah Jane	39 10:37:26 AM	Trip to Washing	09/27/1999	Taxi	\$15.00
0	0	Sarah Jane	39 11:38:20 AM	Lunch with #1 c	09/30/1999	Meal	\$55.75
0	0	Sarah Jane	39 11:39 14 AM	Lunch with #1 c	09/30/1999	Tips	\$8.00
0	0	Sarah Jane	399 5:24:22 PM	Conference in C	10/06/1999	Air	\$249.00
0	0	Sarah Jane	399 5:25:04 PM	Conference in C	10/06/1999	Meal	\$22.50
0	0	No one	39 10:50:19 AM				

Figure 2-12: Viewing records that have been uploaded to the PC

Note that in addition to the fields on your form, there are four additional database columns: RecordID, UnitID, UserName, and TimeStamp. These four fields are generated every time you create a new record. Pendragon Forms uses the UnitID, UserName, and TimeStamp to uniquely identify individual records. The UserName is your Palm user name. By default, only records with your UserName will be sent to your handheld device. The TimeStamp is the creation date and time of the record. If you are keeping records on the handheld for a specific number of days, the TimeStamp is used to determine when a record should be removed.



See Chapter 8, "Advanced Form Properties," for information on sending records to the Palm organizer independent of UserName.

Bi-directional Synchronization

Pendragon Forms supports bi-directional synchronization, meaning that during the HotSync process, records from the handheld will be uploaded to the PC, and records on the PC will be downloaded to the handheld.

When you are viewing data on the PC, you can therefore choose to enter new records or modify existing records.

If you create a new record on the PC, you need to select your UserName for the record to be sent to your handheld device on the next HotSync data transfer.



Note that when you modify existing records, the synchronization rule is that if both the handheld and the PC modify the same record, the changes on the handheld will overwrite the changes on the PC.

Making Changes to a Frozen Form

While you are in the process of designing and testing a form, you are likely to want to make changes to your form design. Once a form design has been frozen, there are some aspects of the form design that you can change, and some that you cannot.

Items that can be changed on a frozen form are:

- ◆ Form Properties, such as Data Persistence options for determining which records are removed from the handheld
- ◆ Advanced Form Properties
- ◆ Advanced Field Properties and Scripts
- ◆ Field names
- ◆ Selection of which Lookup List is referenced in a Lookup List field
- ◆ Selection of which form will be used as a subform

If you make a change to a frozen form, you need to click the name of the form and click the Distribute button to send the changes to the handheld during the next HotSync data transfer.

Items that cannot be changed on a frozen form are:

- ◆ Addition or deletion of fields
- ◆ Items in a Popup List or Multi-Selection List field
- ◆ The field type used in a given field
- ◆ Database column names

If you need to change one of these, you can copy the form design and modify the copy.

Copying a Form Design

You may want to copy a form design if you want to use one form as the starting point for the design of a different form, or if you want to make changes to a form that has been frozen.

When you copy a form, the copy is not frozen. Also, any data in the original form is not transferred into the copy.

To copy a form:

1. In the Pendragon Forms Manager, click the name of a form, and then click the Copy button.

2. The copied form will have the same name as the original, plus an asterisk (*) at the end. You can edit the form to change its name, and then edit fields.
3. When the copy is created, the database column names will remain the same as in the original form design. This is to facilitate importing data from the original form into the copy.



If you have difficulty freezing the form design of a copied form, refer to Appendix A, "Troubleshooting Tips."

Printing a Form Design

While you are designing a form, it may be useful to print the form design so that you can see the order of the fields on the form.

Pendragon Forms has three report options for printing form designs:

- ◆ The Form Design Report displays nine fields per page.
- ◆ The Detailed Form Design Report displays two fields per page and includes scripts. This option is very paper-intensive, and you may want to set your printer properties to print two pages per physical page to reduce the amount of paper that you use.
- ◆ The Screen Preview Report displays four fields per page and shows a sample of what the handheld looks like in Field View for each field.

To print a form design, click the name of a form. Click the File menu, and then select a report option. Press Ctrl+P to print the report.

Summary

This chapter gave you an overview of how Pendragon Forms works. Forms are created on the PC and sent to the Palm organizer, where you can fill in the forms. When you perform a HotSync data transfer, your data is sent back to the PC, where it can be viewed.

Chapter 3

Entering Data on the Palm Organizer

IN THIS CHAPTER

- ◆ Entering new records on the handheld
- ◆ Reviewing existing records
- ◆ Using the Quick Sum feature to add a column across records
- ◆ Filtering and Sorting records
- ◆ Identifying problem records that did not upload to the PC
- ◆ Printing records from the handheld

EACH TIME YOU FILL out a form, you create a new record in the database on the handheld. The record stores the responses that you or your user enters for each field on the form.

Entering New Records on the Handheld

When you tap the Forms icon on the Applications screen of the handheld, the Pendragon Forms program runs. The Forms List, shown in Figure 3-1, displays an alphabetical list of forms that you have sent to the handheld.

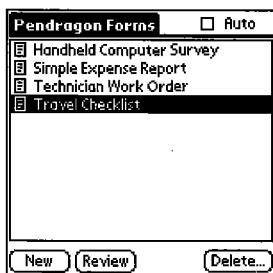


Figure 3-1: The Forms List screen

To create a new record, tap the name of a form to select it, and then tap the New button.

Working in Field View

The default display mode for new records is Field View. As seen in Figure 3-2, Field View displays one field on the handheld screen at a time. The type of response you can enter depends on what field type was selected for the field when the form was designed.

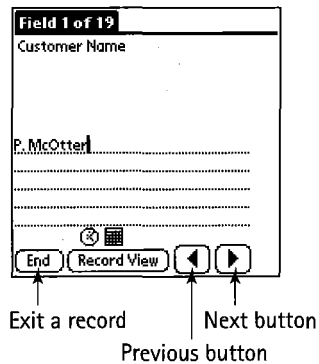


Figure 3-2: Field View on the handheld

After entering a response in a field, tap the Next button to move to the next field on the form. You can step through the form, entering responses and tapping the Next button after each response. If you are on the last field of the form, tapping the Next button will return you to the Forms List.

While entering responses on a form, you can tap the Previous button to move back to verify or correct an earlier response.

The End button enables you to exit a record at any point. If you start a new record and immediately tap the End button before entering any data, the program assumes that you do not want to keep a blank record. As shown in Figure 3-3, a message will ask you to verify that the blank record should be deleted. You can tap Yes to delete the record or No to keep the blank record. Each record contains a response for each field on the form. A blank record therefore contains a blank response for each field.

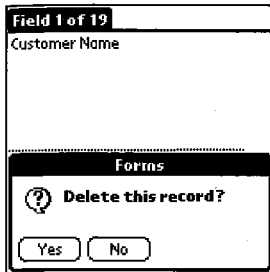


Figure 3-3: The Deletion message is displayed when you exit a blank record.

Using AutoNavigate

The Forms List has an AutoNavigate check box, labeled Auto, which is shown switched on in Figure 3-4.

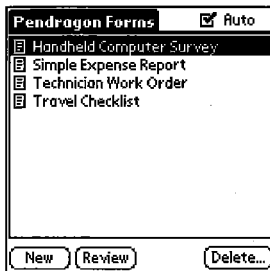


Figure 3-4: AutoNavigate switched on

The AutoNavigate option, when switched on, speeds up data entry in Field View in two ways:

- ◆ When you make a selection in a selection field (such as a Date field, a Popup List, a Yes/No check box, Option 1 of 5, or a Lookup List), AutoNavigate automatically advances to the next field on the form without the handheld user's having to tap the Next button.
- ◆ If the next field on the form is also a selection field, AutoNavigate will automatically display the selection list.

In addition to working with selection fields, AutoNavigate is also useful for advancing to the next field on a form after a bar code has been scanned into a Text field. (To use Pendragon Forms with a bar code scanner, either a Symbol Technologies SPT 1500 handheld is required, or an external bar code scanner can be attached to the Palm organizer. See Chapter 12, "Using Bar Codes.")



If you installed the evaluation version of Pendragon Forms from the CD-ROM, open the Pendragon Forms Manager. In the Category field, select the Samples category. Distribute the Handheld Computer Survey form to your handheld. To compare the speed of data entry, fill out a new record with AutoNavigate switched on, and then fill out a new record with AutoNavigate switched off.



One of the drawbacks of the AutoNavigate feature is that if the next field on a form is a Date field or a Lookup List, the display of the calendar or list fills the entire screen and obscures the field name. If you or your handheld user needs to read the field name to obtain instructions, then you may prefer to switch off AutoNavigate. You can set an Advanced Form Property of Disable AutoNavigate — see Chapter 8, "Advanced Form Properties."

AutoNavigate applies only when you are creating a new record in Field View. If you are reviewing an existing record or creating a new record in Record View, turning on AutoNavigate has no effect.

When AutoNavigate is switched on, it applies to all new records in all form designs, except if a form has the Advanced Form Property of Disable AutoNavigate switched on.



For information on Advanced Form Properties, see Chapter 8.

Working in Record View

In addition to Field View, Pendragon Forms offers a second way to display a form on the handheld: Record View. As shown in Figure 3-5, Record View displays a form in a two-column format, with Field Names in the left-hand column and Responses in the right-hand column.

Figure 3-5: Record View on the handheld

If you are in Field View, as illustrated on the left in Figure 3-6, tap the Record View button to switch to Record View. As seen on the right side of the figure, the field that was being displayed in Field View will be at the top of the screen in Record View.

Figure 3-6: In Field View, tapping the Record View button switches you to Record View; the field that was displayed in Field View appears at the top of the Record View screen.

Record View displays eleven fields at a time. To enter a response for a given field, tap in the response area of that field.

You have two ways to traverse a form in Record View:

- ◆ The up and down arrow buttons at the bottom of the screen enable you to move up or down one field at a time.
- ◆ The Page Up and Page Down buttons on the handheld (conventionally called the hardware buttons) enable you to move up or down a screen (ten fields) at a time.

In addition to providing you with the ability to move up and down the fields of a single record on a form, Record View also displays right and left arrow buttons that enable you to move from one record to the next on the same form. The End button enables you to exit a record and return to the Forms List screen.

Three Faces of Record View

Record View actually has three modes. The Normal mode, shown in the figure on the left, is the two-column format that assigns equal screen real estate to the Field Names (left-hand) column and to the Responses (right-hand) column.

If you do not need to give your field names much space, you can choose Wide mode. As shown in the figure on the right, Wide mode makes the Responses column wider than the Field Names column.

James Lewiston	
Customer Name	James Lewiston.....
Customer Address	185 East Maple Cre.....
Date of Visit	6/15/99
Start Time	<input checked="" type="checkbox"/>
Type of Work perf	Repair
Installation	
Item Installed	Washing machine a.....
Old item removed	<input checked="" type="checkbox"/>
Extra parts?	<input type="checkbox"/>
Repair Work	
Item to Repair:
End <input type="button" value="Up"/> <input type="button" value="Left"/> <input type="button" value="Right"/> <input type="button" value="Down"/>	

Normal Record View

James Lewiston	
Customer N	James Lewiston.....
Customer A	185 East Maple Crescent.....
Date of Visit	6/15/99
Start Time	<input checked="" type="checkbox"/>
Type of Wo	Repair
Installation	
Item Install	Washing machine and gas.....
Old item re	<input checked="" type="checkbox"/>
Extra parts	<input type="checkbox"/>
Repair Work	
Item to Rep
End <input type="button" value="Up"/> <input type="button" value="Left"/> <input type="button" value="Right"/> <input type="button" value="Down"/>	

Wide Record View

If you have a very long form and you want to locate yourself using the field number, you can switch on Field Numbers in Record View, as shown in the following figure.

James Lewiston	
1	Customer Name James Lewiston
2	Customer Address 185 East Maple Ct
3	Date of Visit 6/15/99
4	Start Time <input checked="" type="checkbox"/>
5	Type of Work p <input checked="" type="checkbox"/> Repair
6	Installation
7	Item Installed Washing machin.
8	Old item remov <input checked="" type="checkbox"/>
9	Extra parts? <input type="checkbox"/>
10	Repair Work
11	Item to Repair:

End

Record View with Field Numbers switched on

To change the Record View mode being displayed:

1. In Record View, tap the handheld Menu button and then select the Option menu.
2. Select Toggle Wide Mode to switch on Wide mode, or select Toggle Field Num to switch on Field Numbers. These menu options are toggle switches, meaning that if you select the same option again, you will switch that option off.

You can also use the Graffiti shortcut /W for Wide mode, and /F for Field Numbers.

It is possible to have both Wide mode and Field Numbers switched on simultaneously, but doing so severely truncates the field names.



Some form designs are easier to use in Record View than in Field View. If you installed the evaluation version of Pendragon Forms from the CD-ROM, open the Pendragon Forms Manager and select the Samples category. Distribute the form Travel Checklist to your handheld, and compare entering a new record in Field View with doing so in Record View.

Jumping from Record View to Field View and Back

Record View is the fastest way to enter data if your form design has a lot of selection fields, especially Yes/No check boxes. However, if you have Text fields on your form, or if you have field names with a lot of instructions, Record View may not display enough characters to suit your needs.

A simple solution is to switch back to Field View whenever you need to view an entire field name or to enter a lot of text in a Text field. You can then switch back to Record View after filling in the selected field.

To switch to Field View, tap the name of a field in the left-hand column in Record View, as shown on the left in Figure 3-7. The field that you selected will be displayed in Field View, as shown on the right.

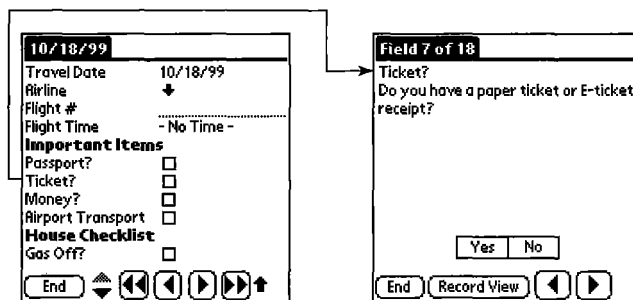


Figure 3-7: Tap the name of a field in Record View to switch to Field View. The selected field will be displayed in Field View.

To switch from Field View back to Record View, tap the Record View button.

Changing the Default View for New Records

Field View is the default view for new records created on the handheld. If you prefer to enter new records in Record View, you can change the default view globally across all forms.

From the Forms List, tap the handheld Menu button. On the Options menu, select Toggle Default View. (You can also use the Graffiti shortcut /V.) This changes the default view for entering new records from Field View to Record View.

To switch back to Field View as the default, tap Toggle Default View again.

If you do not want to set the default view globally across all form designs but instead prefer to choose between Field View and Record View on a form-by-form basis, refer to Chapter 8, “Advanced Form Properties.”

Reviewing Records on the Handheld

If a record already exists on the handheld, then the way to access the record is via the Review button on the Forms List screen.

You may want to review records on the handheld for several reasons:

- ◆ You may want to refer to or update a record that you previously created on the handheld.
- ◆ You may be populating the form with records from the desktop PC. Such records may be partially filled in on the PC and then sent to the handheld to be completed.

To review records on the handheld, tap the name of a form to select the form, and then tap the Review button.

The Records List screen, shown in Figure 3-8, displays a list of the records that are present on the handheld for the selected form.

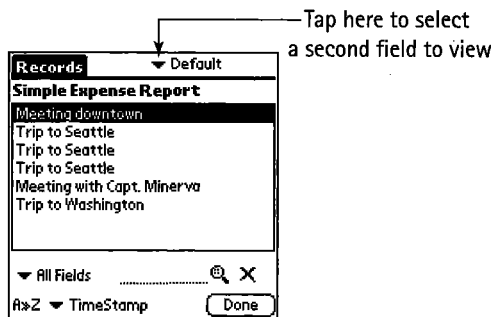


Figure 3-8: The Records List Screen displays the existing records for the selected form.

The Display Key, typically the first field on the form, is used to display the records. Generally, a good form design uses a unique field such as a Text, Numeric, or Date field for the Display Key so that it is easy to distinguish between records on a form.



Refer to Chapter 2, "Creating a Form," for information on selecting an appropriate Display Key field. If you do not want the first field on your form to be the Display Key, see Chapter 6, "Advanced Field Properties."

If the Display Key does not give you enough information to be able to tell two records apart, you can choose to view a second field on the Records List.

To select a second field to view in the Records List screen, tap the arrow in the upper-right corner of the Records List screen, and select another field on the form to view.

In Figure 3-8, the Display Key is a description of an expense, but several expenses have the same description. Therefore, as shown in Figure 3-9, another field, in this case the type of expense, is selected for display to help distinguish which record belongs to which expense.

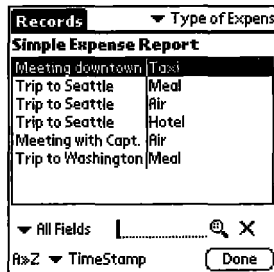


Figure 3-9: The Records List with a second field displayed

Once you can identify which record you want to review, tap the record to select it. The record will be displayed in Record View mode, as shown in Figure 3-10. You can modify the record as necessary, and you can switch between Record View and Field View.

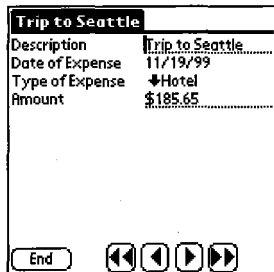


Figure 3-10: Record View mode is used to review records.

If you tap the End button, you will be returned to the Records List screen, as shown in Figure 3-9. If you have no more records to review, tap the Done button on the Review screen to return to the Forms List.

Quick Sum and Average

Quick Sum is a useful function that can be performed from the Records List screen to add up the values of a selected field across all records. For example, if you create a form to log flight miles, and one field on the form is used to record total miles for a given trip, then you can use the Quick Sum feature to add up total miles for all trips.

The Quick Sum feature works as follows: The column to be added is displayed as the second column in the Records List screen, as shown in the figure on the left. Tap the handheld Menu button, and select the View menu, as illustrated in the figure on the right. Choose the Column Total option. You can also use the Graffiti shortcut /T to display the Column Total.

Records		Amount
Simple Expense Report		
Meeting downtown	\$18.00	
Trip to Seattle	\$22.50	
Trip to Seattle	\$350.00	
Trip to Seattle	\$185.65	
Meeting with Capt.	\$195.00	
Trip to Washington	\$35.00	

▼ All Fields [Info] [X]
A-Z ▼ TimeStamp [Done]

Make the column that you want to add the second display column in the Records List screen.

Record View	
Simple	Toggle Record Num ✓/R
Meet	Toggle Change Flags ✓/M
Trip to	Column Total... ✓/T
Trip to Seattle	\$350.00
Trip to Seattle	\$185.65
Meeting with Capt.	\$195.00
Trip to Washington	\$35.00

▼ All Fields [Info] [X]
A-Z ▼ TimeStamp [Done]

Tap the handheld menu to access the View menu, and select the Column Total option.

The Column Total screen, shown in the following figure, displays the sum of values for the selected field across all records, and the average value in this field. The count is the number of records that have been created for this form.

Records		Amount
Simple Expense Report		
Meeting downtown	\$18.00	
Column Total		
[Info]	Total for column 'Amount':	
	SUM: \$806.15	
	AVG: \$134.35	
	COUNT: 6	
[OK]		

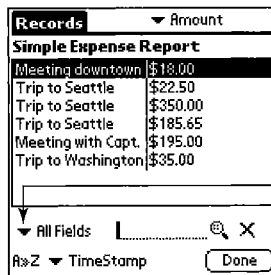
The Column Total screen

Quick Sum works with Numeric and Currency fields. Quick Sum will also work with a Text field if you have entered purely numeric values. Quicksum does not provide a way to copy a sum into a field. If you need to store a total in a record on a form, you can use a Button field with a script to calculate the total (see Chapter 11, "Using Scripts").

Filtering Records

The Filter feature enables you to specify search criteria and display only the records that match the criteria. You can search across all fields on a form, or you can limit your search to one specific field.

The Filter feature is accessed from the Records List screen. As shown in Figure 3-11, the default is to search through all fields. To reduce the search time, you can choose to search in a specific field, as shown in Figure 3-12.



The default is to search all fields on the form

Figure 3-11: The Records List with the default setting to search All Fields

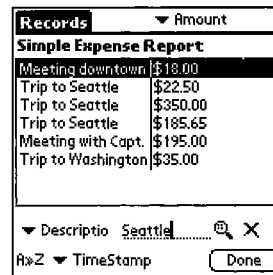
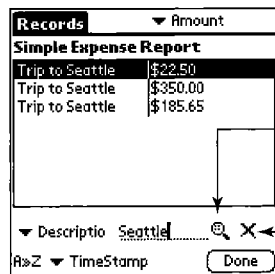


Figure 3-12: You can choose to search a specific field (in this case, Description).

To enter the search criterion, tap in the dotted line and enter a character string. Tap the magnifying glass icon to perform the search. As Figure 3-13 shows, only the records that match the search criterion will be displayed.



Tap the magnifying glass to apply filter

Tap X to switch off filter

Figure 3-13: A filtered list of records

Reviewing Filtered Records

To review a record, tap the record to select it. You can then modify the record as needed. When you end the record by tapping the End button on the record, you will return to the Records List. The filtered list will still be displayed.

To switch off the filter and return to viewing a list of all records for the form, tap the X icon to the right of the magnifying glass.



If you use the Quick Sum feature while a filtered list of records is displayed, you will see the sum of just the filtered records. For example, if you have a form that displays sales orders, you can filter on a customer's name and then use Quick Sum to determine the total sales that the customer has given you.

Filtering Records with an SPT 1500 Handheld

If you are using a Symbol Technologies SPT 1500 handheld, and if your form contains a Text field with a bar code, you can scan a bar code into the search criteria field to find any records that contain the bar code. A filtered list of records will be displayed, and you can tap a record to review it.

If you are using unique bar codes, so that no two records have the same bar code, then when you scan the bar code into the search criteria field, the matching record will automatically be displayed.

Sorting Records

The Sort feature enables you to select a field on which to sort the records. You can choose to sort in ascending order (A to Z) or in descending order (Z to A).



As records are refreshed from the PC during a HotSync, they are placed on the handheld in no specific order. This means that after a HotSync, the records may appear in a random order on the handheld. Instead of re-sorting on the handheld after each HotSync, you can specify your own download sort order for records by changing the Additional Download Criteria (see Chapter 8, "Advanced Form Properties"). When a sort order is specified in the Additional Download Criteria, the records are pre-sorted on the PC before being placed on the handheld.

The Records List screen is where records are sorted. Ideally, you should choose to display the field that you want to sort, so that you can verify that the records sort correctly.

To sort the records on a given field, tap the arrow next to the A>Z icon and select a field to sort by. Records will be sorted in ascending order (A to Z) in that

field. If you prefer to sort in descending order, tap the A>Z icon. The icon will display Z>A and the records will be sorted in descending order.

Figure 3-14 shows records before a sort criterion is applied, and Figure 3-15 shows the result of the sort.

Records	
▼ Date of Visit	
Technician Work Order	
Allene	6/17/99
Richards	6/14/99
Smithe	6/15/99
Jones	6/22/99
Smith	6/17/99
Lewiston	6/15/99

▼ All Fields [Search] [X]

A>Z ▼ TimeStamp [Done]

Tap here to select
a field to sort by

Figure 3-14: Selecting a field to sort by

Records	
▼ Date of Visit	
Technician Work Order	
Richards	6/14/99
Smithe	6/15/99
Lewiston	6/16/99
Smith	6/17/99
Allene	6/17/99
Jones	6/22/99

▼ All Fields [Search] [X]

A>Z ▼ Date of Visit [Done]

Figure 3-15: The records are sorted by the selected field.



If you choose to sort records by the TimeStamp field, you can sort the records in the order in which they were created.

Pendragon Forms does not do a multivariable sort, meaning that if for instance you are displaying Last Name and First Name fields, and you choose to sort on Last Name, records will not also be sorted on First Name.

Sending Records to the PC

Records that are new or that have been modified on the handheld are of special significance because these records will be written to the database on the PC during the next HotSync data transfer. In the case of a new record on the handheld, the record will be added to the database on the PC. In the case of a modified record on the handheld, the changed record will overwrite the matching record on the PC.

Identifying New and Changed Records

The Records List screen, which displays the list of existing records on the handheld, has a convenient Change Flags option that enables you to identify which records are new or have been changed since the last HotSync data transfer.

To switch on the Change Flags option, select a form from the Forms List and then tap the Review button to display the Records List. Tap the handheld Menu button, select the View menu, and then choose the Toggle Change Flags option. (You can also use the Graffiti shortcut /M.) In the Records List, as shown in Figure 3-16, any new or changed records will be marked with an asterisk.

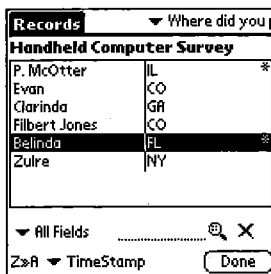


Figure 3-16: When the Change Flags option is switched on, any new or changed records will be marked with an asterisk.

After a HotSync data transfer, the asterisks will be removed because the records have been sent to the PC. Any new records that you create or modify after the HotSync process will be assigned the asterisk flag to show that those records have changed since the last HotSync data transfer.

To switch off viewing the Change Flags, tap the Toggle Change Flags menu option again, or use the Graffiti shortcut /M.



The Records List screen includes a Record menu with an option called Mark All Changed. This option flags every record on the handheld as a changed record to be updated in the database on the PC. If the PC has been updated since your last synchronization, the older records on the handheld will replace the newer records on the PC. In a multiuser environment, it's not a good idea to use the Mark All Changed option, because of the risk of an old record's overwriting a new record.

Identifying Problem Records That Did Not Upload to the PC

During the HotSync process, all new and changed records should be sent to the database on the PC. Under certain circumstances, there may be a problem that causes a record on the handheld not to be uploaded to the PC.

Examples of problem scenarios include:

- ◆ If you fail to fill in a required field on a form, and if you then immediately initiate the HotSync process before ending the record, the record will not be uploaded to the PC because it is missing required information.
- ◆ If you are using a primary key on your form, and a new record on the handheld has the same primary key as an existing record in the database on the PC, the record on the handheld will not be allowed to upload because it will overwrite the older record. (See Chapter 6, “Advanced Field Properties,” for information on primary keys.)

If there is a problem record that cannot be uploaded to the PC, an error message will be generated during the HotSync process to alert you to the fact that a problem has occurred.



See Appendix A, “Troubleshooting Tips,” for a list of HotSync Log error messages.

After the HotSync data transfer, you can easily identify the problem records by tapping the name of the form in the Forms List and then tapping the Review button. As shown in Figure 3-17, records that were not uploaded to the PC are flagged with an arrow.

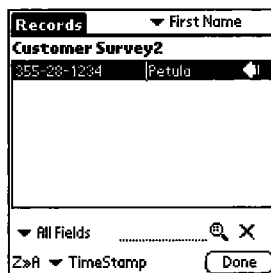


Figure 3-17: Records that are not uploaded to the PC are flagged with an arrow in the Records List.

To fix a problem record, review the record and make necessary changes. For example, fill in required fields or modify the primary key fields to be unique.

Deleting Records and Forms from the Handheld

Pendragon Forms version 3 is designed to be centrally managed from the PC, not the handheld. This means that if you want to remove a form and the records in that form, you will need to remove the form from the User Group to which that form belongs. If you remove the form from the handheld without removing the form from the User Group, the form will be re-sent to the handheld during the next HotSync data transfer.



For instructions on deleting a form or a record, refer to Chapter 7, "Synchronization Rules."

Manually Deleting Forms and Records from the Handheld

Under certain circumstances, you can manually delete individual records or entire form designs and their records.

- ◆ If a record has been sent to the handheld from the PC, and if incorrect information was entered on the handheld, you may choose to manually delete the individual record. Then during the next HotSync data transfer, a fresh copy of the record will be sent to the handheld from the PC.
- ◆ If a form has been deleted from the PC before it was removed from the handheld, the form may be deleted manually if you are certain that you do not need the records in the form. If you need the records, do not delete the form.

To manually delete an individual record or an entire form plus all associated records, tap the name of the form in the Forms List, and then tap the Delete button. As shown in Figure 3-18, a list of records for the form is displayed.

To delete a record, tap the record and then tap the Records button. Tap the Delete button. You will be prompted to confirm the deletion.

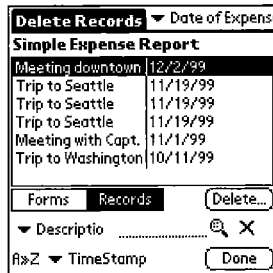


Figure 3-18: The Delete screen on the handheld

To delete a form, tap the Forms button to display a list of forms. Tap a form to select it, and then tap the Delete button. You will need to confirm the deletion. Deleting a form also deletes any records associated with that form.

Limiting Handheld Access Rights

On the PC, the Form Properties window is used to determine how long records will remain on the handheld.

As shown in Figure 3-19, the Form Properties window also contains an Access Rights section that gives the handheld user permission to add new records, update records, or delete records.

To access the Form Properties window, click the name of a form in the Pendragon Forms Manager and then click the Properties button.

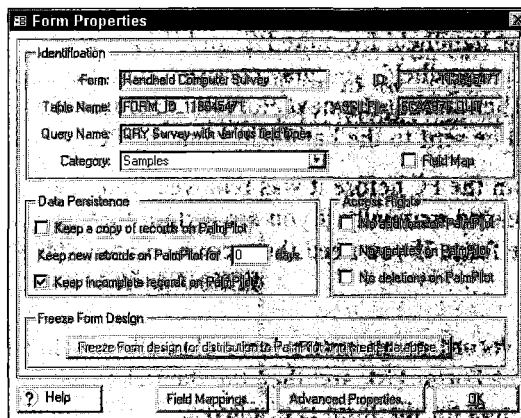


Figure 3-19: The Form Properties window

The possible access rights settings are:

- ◆ **Default setting – No check boxes checked.** This means that the handheld user can add new records, modify existing records, and delete records.
- ◆ **No Additions on Handheld.** If this box is checked, the handheld user will not be able to create new records. This option may be useful in a work order scenario in which new work orders are always created on the PC and sent to the handheld.
- ◆ **No Updates on Handheld.** If this box is checked, the handheld user cannot modify existing records. This may be useful if you are sending reference information to the handheld that you do not want the handheld user to modify.
- ◆ **No Deletions on Handheld.** If this box is checked, the handheld user cannot delete records from within the Pendragon Forms application on the handheld. To prevent deletion of a form design from within the Forms application, refer to Chapter 8, “Advanced Form Properties.”

Protecting deletion of forms and records from within the Pendragon Forms application is not a foolproof method. There is still a risk that the handheld user can delete the entire Forms application, along with all form designs and data, from the Delete menu on the Palm Applications screen.



If you set an access right and then change your mind, simply change the access right and then redistribute the form to the handheld. After a HotSync data transfer, the handheld will have the updated access rights for the form.



If you delete a form design from the Pendragon Forms Manager, you will lose the ability to change the access rights, even if the form is still present on the handheld. As a rule, never delete a form design from the PC unless you have removed the form from the handheld as well. See Chapter 7, “Synchronization Rules,” for information on deleting forms.

Printing from the Handheld

There are two ways to print directly from the handheld:

- ◆ Via the infrared port to a printer with infrared capability
- ◆ Via the handheld serial port to a serial printer

Infrared Printing

Infrared printing requires external software to be installed on the handheld. Pendragon Forms supports Bachmann Print Manager software version 2.0 or later.

Bachmann Print Manager supports printing to most infrared-capable printers such as the Hewlett-Packard DeskJet 340Cbi and the Canon BJC-50.



An evaluation copy of Bachmann Print Manager software is on the CD-ROM.

Serial Port Printing

Serial port printing requires a serial RS-232C port printer. Pendragon Forms supports printing to the Seiko Instruments Inc. DPU-201GS Thermal Printer and the DPU-414 printer. However, if your serial printer will accept ASCII data at 9600-N-8-1, there's a good chance that it will work with the software.

Additional cables are required to connect the handheld to the printer. If you use the HotSync cradle or HotSync cable, you will need a 9-pin Male to 9-pin Male Gender Changer adapter, and a 9-pin Male to 9-pin Female Null Modem adapter. The Null Modem adapter is required because the HotSync cable or cradle has a built-in null modem adapter that has to be canceled out.

Another possible cable connection to the Palm organizer is the Palm modem cable, which is a straight-through cable without a null modem adapter. If you use the Palm modem cable, you will need a 25-pin Female to 9-pin Male adapter, just like the one supplied with the standard HotSync cradle.

Printing a Record

There are two ways to print from the handheld:

- ◆ **Printing via a Button field.** If a button field has been scripted to be a Print button, you can tap the Print button to print the current record. Refer to Chapter 11, "Using Scripts," for information on a Print button script.

- ◆ **Printing in Record View.** If a record is displayed in Record View, you can tap the handheld Menu button, select the Record menu, and tap the Print option.

Printer Output

Pendragon Forms prints all fields on the form by default. The printer output is very basic: a field name on one line, followed by the response to that field on the next line, followed by the next field name, and then the next response, and so on.

Signature fields print out as the word “Signed” or “Unsigned.”

It is not currently possible to customize print output, although you can choose whether or not to print a given field. To select which fields are nonprinting, refer to Chapter 6, “Advanced Field Properties.”

Summary

This chapter illustrated the basics of using Pendragon Forms on the Palm organizer. You can now create new records on the handheld or review, filter, and sort existing records. You can also switch between the display modes of Field View and Record View on the handheld.

Part III

Form Design Techniques

IN THIS PART

CHAPTER 4

Planning a Form Design

CHAPTER 5

Field Types

CHAPTER 6

Advanced Field Properties

CHAPTER 7

Synchronization Rules

CHAPTER 8

Advanced Form Properties

Chapter 4

Planning a Form Design

IN THIS CHAPTER

- ◆ Making forms easy to use
- ◆ Optimizing performance on the handheld
- ◆ Optimizing memory usage
- ◆ Analyzing the frequency and performance of synchronization

Planning a Form

You have several key questions to ask when designing a data collection system.

- ◆ Who will be using the form?
- ◆ How many records are created per day?
- ◆ How much data will a Palm organizer hold?
- ◆ How will users synchronize their handhelds?

The answers to these questions will help you to identify the constraints on forms that you can realistically design and implement.

The following sections cover some common issues and techniques that relate to form usability, performance, memory usage, and synchronization.

Form Usability

Making your form easy to use will improve data entry accuracy and enhance the user experience. This is especially important when you create a form to be shared by your workgroup. If other workgroup members have difficulty using the form, they will be less likely to use it and may even grow to resent using it.

Though you can build a form with Pendragon Forms in just five minutes, it may take some time to refine your design to make it easier to use.

Tips for Designing Forms

These few simple tips will go a long way toward making your forms more user-friendly.

- ◆ Avoid thinking too deeply about the desktop component of your application, and focus instead on the handheld form design. In a multiuser scenario, it is best to simplify the application on the handheld, even at the expense of complexity on the desktop. The most important attribute of your application is that it is mobile; if the mobile piece doesn't work, there's little point in building the application.
- ◆ Minimize the amount of text the user must enter with the stylus. You can do this by replacing a Text field with a Popup List or Lookup List field. For example, if you have a Text field on your form for recording the weather conditions, you can probably replace this field with a Popup List field that covers 95 percent of possible conditions. A second Text field could be used when the user selects "Other" from the list.



In Chapter 11, "Using Scripts," you will learn how to use scripts to automatically hide and show fields as needed.

- ◆ Another way to reduce the need for manual data entry is to use default values. If a field can be preset so that the user needs to change it only 10 or 20 percent of the time, you can save quite a lot of data entry effort. For forms that document repetitive tasks, such as inventory control, you can use the AutoDefault property to default a field to its last entered value. For more information on default value settings, see Chapter 6, "Advanced Field Properties."
- ◆ If the user is likely to enter data into fields in order, then Field View usually provides the best interface. Field View has a larger display for long, complex questions, and the interface is simpler, but it doesn't display as much data as Record View.

Performance

Palm Computing handheld devices may appear to be fast computers, but they possess less than one percent of the power of today's desktop PCs. And because handheld computers typically synchronize via the PC serial port, they can transfer data

only at a slow 5KB per second. This means that care must be taken when designing applications for Palm Computing devices, or the device and its synchronization mechanism will be overwhelmed.



Each byte is equivalent to a character of text. One kilobyte (KB) is 1,024 bytes. One megabyte (MB) is 1,024KB, or 1,048,576 bytes.

The key to good performance lies in limiting the amount of data on the handheld. Maintaining fewer records implies faster sort and filter times. It also means shorter synchronization times. As a rule, only the data that the user is likely to need should be loaded onto the handheld.

How Performance Depends on Your Data

The time for various operations depends directly on the amount of data on the handheld.

DATA ENTRY

Data entry speed will slowly be reduced as more records are added into memory. This slowdown is more apparent if you have created your own unique, primary keys for your form. This is because the software must check all other records for duplicate keys before saving records. This delay is proportional to the number of records.

SORTING

The time needed to sort records increases rapidly with the number of records. If you double the number of records, the sort could take three to four times longer.

FILTERING

If you filter on a specific field, the search time is proportional to the number of records. If you filter on All Fields, the search time is proportional to the number of records and will also increase with the number of fields.

SYNCHRONIZATION

Synchronization time is generally proportional to the total amount of data that is transferred, but an overhead is associated with each record. It is somewhat less efficient to transfer lots of small records than it is to transfer a few large records, even if the total number of bytes transferred is the same.

Estimating Memory Usage

For many applications, only a few records are required to be present on the handheld at any time. For instance, an inspector who will perform one or two

inspections before returning to the office to transfer data from the handheld to the desktop will use only a minimal amount of memory. In such cases, there's no need to worry about memory usage.

On the other hand, if you're not certain how many records are likely to be collected in the field, or if you are sending a large number of records from the desktop to the handheld, you should do a rough calculation to determine the memory requirements for your application.

RECORDS

The following formula can be used to estimate the size of an individual record:

64 bytes + $N \times 28$ bytes + the greater of (the size of the actual data in each field or 14 bytes per field)

where N is the number of fields on the form. Note that the number of fields on a form is limited to 250.

This means that a ten-field form with all Text fields will take up at least:

64 bytes + 10×28 bytes + $10 \times 14 = 484$ bytes

Of course, Text fields can take up more than 14 bytes. The number of bytes used for the data in a Text field is equal to the number of characters in the Text field plus one, rounded up to the next even number of bytes. If a handheld user enters 25 characters in a Text field, for example, that field will take up 26 bytes.

The default maximum size of a Text field is 255 characters. However, you can set the maximum size of a Text field to 2,000 characters. If Text fields were all filled to the maximum capacity, each field would occupy 2,002 bytes. On a form consisting of ten such Text fields, the size of such a record would be:

64 bytes + 10×28 bytes + $10 \times 2002 = 20364$ bytes

As you can see, the size of each record can vary greatly depending on its contents, and this in turn will depend on the particular application.



Note that the maximum size of a record is about 64,000 bytes. If you enter data into a field that will push the record size over this limit, you will receive a message on the handheld warning you that the record could not be saved.

If possible, perform an experiment: Check the memory usage on your handheld before and after loading your form with about 25 sample records.

As of this writing, all Palm Computing devices that are in production have at least 2MB (2,048KB) of memory. Some of this memory may be used by other programs and data, such as the built-in Address Book and Date Book applications. After excluding about 300KB for these other applications and 160KB for the

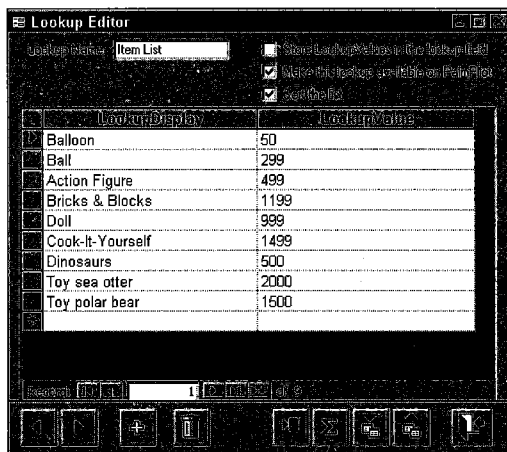
Pendragon Forms program itself, about 1,500KB remains for all of your record data on a 2MB device.

For data collection applications, you will need to estimate the maximum number of records likely to be collected on the handheld each day. The larger the size of each record, the fewer records you will be able to enter before synchronizing and dumping the records to the PC. For example, a form with 150 fields might take up 10KB for each record, limiting you to about 150 records on a 2MB device. However, this is not as large an issue as it would seem. This is because the total number of fields that can be entered works out roughly the same, whether you enter 150 records with 150 fields or 1,500 records with 15 fields. In both cases, you end up entering about 22,500 fields before you fill up memory and need to synchronize.

For applications in which data will be maintained on the handheld, the most important issue is the number of records because this limits the performance of the system.

LOOKUP LISTS

Lookup Lists don't take up much more space than the text in the list itself. To find out how much memory will be used by your lookup lists, you can use the Statistics button in the Lookup List Editor, shown in Figure 4-1, to get an estimate.



Statistics button

Figure 4-1: The Lookup Editor window and the Statistics button

When you click the Statistics button, a list displays the approximate memory usage of each Lookup List. As shown in Figure 4-2, the memory usage is displayed in bytes.

Lookup Name	Used On Pilot	Display Memory Usage	Value Memory Usage	Total Memory Usage
HardwareHard D	<input checked="" type="checkbox"/>	39	4	71
HardwareMonitor	<input checked="" type="checkbox"/>	25	4	57
HardwarePrinter	<input checked="" type="checkbox"/>	48	6	80
Item List	<input checked="" type="checkbox"/>	104	39	136
Price List	<input checked="" type="checkbox"/>	104	39	136
States	<input checked="" type="checkbox"/>	487	153	519
Installation Items	<input type="checkbox"/>	16	1	48

Figure 4-2: Lookup List memory usage

If you are no longer using a Lookup List on the handheld, you can save memory on the handheld by unchecking the check box labeled Make This Lookup Available on Handheld. The Lookup List will not be sent to the handheld during a HotSync data transfer.

FORM DESIGNS

Form designs generally take up one to two times as much memory as the corresponding record. If you have long scripts, pop-up lists, or field names, the form design may be larger.

Reducing the Amount of Data on the Handheld

Several mechanisms can control the number of records on the handheld. The basic methods are to remove all records after synchronization, keep records on the handheld for a specific number of days, or keep the records on the handheld until marked as complete. These settings correspond to the Data Persistence settings described in Chapter 2, “Creating a Form.”

You can fine-tune the basic Data Persistence features by further limiting the data with the Advanced Form Property settings. These settings will give you the ability to limit the number of records maintained on the handheld in some interesting ways.

- ◆ The total memory usage option limits the number of records used by a form by the total volume of memory they take up. A setting of 200KB is equivalent to saying “place as many records on the handheld as will fit in 200KB.”

Checking How Much Free Memory Is on the Handheld

To check how much free memory you have on your handheld, tap the Applications button and then tap the Menu button. Select the Info menu option.

As shown in the accompanying figure, an Info screen will display how much memory is being used by each application, and how much free memory remains.



The Info screen on the handheld displays how much free memory is available.

- ◆ An archive date setting may be used to exclude all records entered before a fixed date.
- ◆ By using a SQL condition, you can restrict records by priority, age, status, or almost any other criteria. SQL is a database query language used by relational databases. The SQL language is an advanced topic that is beyond the scope of this book, but some simple examples of SQL conditions are covered in Chapter 8, “Advanced Form Properties.”

The Advanced Form Properties are discussed in greater detail in Chapter 8. You can change the Advanced Form Property Settings at any time, even after a form design has been frozen. If a form is already on the handheld and you change the Advanced Form Property settings, you will need to click the name of the form in the Pendragon Forms Manager, and click the Distribute button to redistribute the form. Perform a HotSync data transfer to send the updated form design to the handheld.

How Often Should You Synchronize?

The short answer to the question of how often you should synchronize is that the handheld user should do so once per day. The longer you postpone synchronization,

the greater the risk that you will lose data because of loss of battery power, mechanical damage to the handheld, or other mishaps.

If you are collecting a large number of records, or if the records are critically important, you should consider synchronizing more than once per day. In the case of a 200-field inspection form, for example, you may want to synchronize after each inspection.

Always be aware of the consequences of losing your data, and take precautions to back up your information as often as possible.

Using the printing feature of Pendragon Forms may be a way to provide an on-site backup of your data. See Chapter 3, "Entering Data on the Palm Organizer," for more information on printing.

Synchronization Speed

The standard desktop synchronization speed via the HotSync Manager software is close to 56 Kbps, or about 5KB per second. (1 Kbps is equal to 1024 bits per second.) Some serial ports are capable of 115 Kbps or about 10KB per second, and Palm OS 3.3 will support this setting. Because some of the data transferred is used simply for managing the connection, the effective transmission speed is typically smaller than the numbers quoted here.

Modem HotSync Transfers

If you will be synchronizing via a modem attached to the handheld, the data transfer rate could be considerably smaller. First, the modem itself may be limited in speed. The PalmModem clip-on modem for the Palm III and Palm IIIx is limited to a mere 14.4 Kbps to save battery life. The Palm V clip-on modem has a maximum transfer rate of 36.6 Kbps, and this assumes a good telephone connection. If you are synchronizing via modem, expect synchronization to take considerably longer.

Network HotSync Transfers

The Network HotSync mechanism enables a Palm Computing device to synchronize to one PC by connecting to the serial port of another PC on the same network. The idea here is that you can synchronize with the PC at your desk by synchronizing at any other PC in your company that is on the same network. Both PCs must be running the Network HotSync software, which is available from 3Com.

The Network HotSync Manager synchronizes via TCP/IP network protocol – the same protocol used for Internet communications. Because Palm devices can connect to a TCP/IP network via a modem, you can synchronize in two more ways:

- ◆ By dialing into a private TCP/IP network (such as your company network), you can synchronize with a desktop PC remotely. This approach is relatively secure.
- ◆ If your primary PC is connected to the Internet, you can synchronize via the Internet by dialing into a public Internet service provider (ISP). Synchronizing this way is not secure, but it may be less expensive if your ISP is a national provider and has local phone numbers all over the country.

Server Synchronization

If you have multiple users who will need to synchronize by dialing into a network, the HotSync Manager software is unlikely to provide the best solution. This is because the HotSync Manager software allows only one user to synchronize at a time. Once you have more than a few users out in the field, the chances are that two or more users will attempt to synchronize at the same time. In this case, the second user will get a busy signal (Modem HotSync) or will be placed in a queue for synchronization (Network HotSync). Remote synchronization is slow under almost any circumstances, so getting busy signals or synchronization delays will add to the frustration of remote users. In addition, the added dial-in attempts and slowdowns will use extra battery power. The only way to solve this problem with the HotSync Manager software is to buy additional PCs with modems or network cards.

The elegant solution is to use a synchronization server. With server synchronization software, you can have multiple Palm Computing devices simultaneously synchronize with a single server via TCP/IP. The WaveSync server from WaveWare Communications is a server-oriented replacement for the HotSync Manager software, and it supports Pendragon Forms. For applications where users will need to synchronize via modem or wireless connection, the WaveSync software can significantly accelerate data transfers via TCP/IP, and it permits multiple simultaneous connections. It also enables the administrator to manage multiple users from a single location.



An evaluation version of the WaveSync synchronization server is available on the CD-ROM.



For more information on issues involved in a multiuser environment, see Chapter 15, "Planning a Multi-User Installation."

Tips for Improving Synchronization Performance

Here are a few simple tips to aid in optimizing synchronization of your handheld and PC.

- ◆ Click the HotSync Manager icon in the Task Tray and choose Custom. In the Custom Settings dialog, you can disable any conduits that don't need to be synchronized remotely. For example, if your field workers don't need to use the Memo Pad application, you can disable the Memo Pad synchronization. To disable a conduit indefinitely, you have to choose to set *Do Nothing* as the default setting.
- ◆ If possible, perform your initial two synchronization sessions at a PC. This will permit the PC to install and back up applications and data while still using a fast connection.
- ◆ The WaveSync server from WaveWare Communications can significantly accelerate data transfers via modem. For more information on the WaveSync server, see Chapter 15, "Planning a Multi-User Installation."

Summary

In this chapter you learned how the handheld's memory, frequency of synchronization, and the synchronization speed all have to be considered when you are designing a form.

Chapter 5

Field Types

IN THIS CHAPTER

- ◆ Using general-purpose fields for data collection
- ◆ Selecting fields for entering numbers
- ◆ Minimizing handheld data entry with selection fields
- ◆ Using fields for dates and times
- ◆ Creating navigation fields
- ◆ Using fields to access related forms
- ◆ Customizing a form with specialty fields

IN PENDRAGON FORMS, field types are key in creating a form that is usable on the handheld and whose data is analyzable on the PC.

As you design a form, each field on the form has a name and a field type. The field name gives the handheld user information and instructions on the type of data that should be entered on the handheld. Using the right field type ensures that the handheld user can enter data only in the appropriate format.



You can have a maximum of 250 fields on a form.

Pendragon Forms' 21 Field Types

Pendragon Forms uses 21 different field types, to give you a variety of data entry inputs on the handheld. The field type that you select for a given field is determined by the type of information that you want to collect on the handheld and your requirements for analyzing the data on the PC after a HotSync data transfer.

The 21 different field types are clustered in Table 5-1 in groups of similar functions. Subsequently each field type is described in detail.

TABLE 5-1 FIELD TYPES

Category	Available Field Types
General Purpose	Text field
Fields for Numbers	Numeric Currency
Selection Fields	Yes or No Checkbox Option 1 of 5 Popup List Multi-Selection List Lookup List Exclusive Lookup List
Fields for Dates and Times	Date & Time Date Only Time Time Checkbox
Navigation Fields	Section Jump Popup
Fields to Access Related Forms	Subform List Single Subform
Specialty Fields	Signature Button Read-Only Text Completion Checkbox

Freeform Text Field

The Freeform Text field, also called a Text field, is the most flexible of all field types, in that it can store any characters that are entered on the handheld (see Figure 5-1).

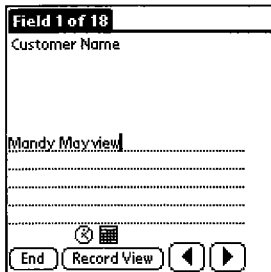


Figure 5-1: A Freeform Text field on the handheld

Text fields are used so frequently that when you create a new field on a form, the field type defaults to Freeform Text. The following table provides an overview of Text fields.

Limits	The maximum length of a Text field is 255 characters.
Alternatives	You can set the maximum number of characters to 2000 by setting the Max Length Advanced Field Property (see Chapter 6).
Handheld Data Entry	Use Graffiti or the handheld keyboard to enter any characters. Tap the calendar icon to add the current date to the field. Tap the clock icon to add the current date and time to the field.
Tips	If you tap the handheld Menu button and choose Lookup, you can perform a lookup to the Palm Address Book and copy a name and phone number into a Text field. You can also use the Graffiti shortcut /L to perform a lookup to the Address Book. With the Advanced Field Property of Lookup List, you can select an item from a Lookup List and paste the selection into the Text fields. For example, some physicians use this feature to speed up writing long phrases. (See Chapter 6.) Text fields can accept data from a GPS (Global Positioning Satellite) receiver. This feature is not widely documented because a GPS receiver can take 5 to 10 minutes to lock on to the satellites, and this is a big battery drain on the handheld. To learn more, refer to the online Help for Pendragon Forms, under the topic of GPS Receivers.

Continued

Continued

Recommended	<p>Use a Text field to enter names, addresses, comments – the sort of data which is variable and unpredictable.</p> <p>Use a Text field to scan a bar code with an SPT 1500 device. Text fields are the only fields that can accept bar code input. See Chapter 12, "Using Bar Codes."</p> <p>Use a Text field to store the results of calculations. You can use the Advanced Field Property of Read-Only to protect the calculated results so that the handheld user cannot modify the results. See Chapter 6, "Advanced Field Properties."</p>
Not Recommended	<p>Using a Text field as a field from which you branch, depending on the data entered, is not recommended. The freeform nature of the field means that the handheld user can enter anything, and this makes it hard to write a branching script that depends on the user input. A Selection field is better for use with branching scripts.</p>
Works with AutoNavigate?	No

Numeric Field

A Numeric field (see Figure 5-2) is a general-purpose field for numeric characters only. The following table describes the specifications of Numeric fields.

Field 6 of 18

How many of our products have you purchased in the last year?

5

+ -	7	8	9
Del	4	5	6
	1	2	3
	0	.	

Enter a number

End Record View ◀ ▶

Figure 5-2: A Numeric field on the handheld

Limits	<p>The maximum length of a Numeric field is 15 digits.</p> <p>A decimal point and a minus sign can be added in addition to the 15 digits.</p>
Alternatives	<p>If you need to enter a number with more than 15 digits, switch to entering numbers as exponents.</p> <p>For example:</p> <p>1,500,000 can be entered as 1.5e6. (1.5e6 means 1.5 multiplied by 10 to the sixth power – that is, $1.5 \times 1,000,000$.)</p> <p>0.0025 can be entered as 2.5e-3</p> <p>The maximum exponent for a positive number is e308, and that for a negative number is e-324.</p>
Handheld Data Entry	<p>In Field View, Pendragon Forms displays a custom numeric keypad that is visible on screen to enter data, or you can use Graffiti.</p> <p>In Record View, use Graffiti.</p> <p>If you pop up the handheld's Numeric keypad, you can enter numeric characters only. You can actually enter nonnumeric characters, but when you try to leave the field, a message will prompt you that the number is invalid.</p> <p>When using exponents, use Graffiti to enter the letter e.</p>
Tips	<p>You can require the handheld user to enter a whole number only, by setting the Integer Advanced Field Property. You might want to use integers if the user is recording a number that cannot have a fractional value, such as the number of customers.</p> <p>You can also specify a numeric range, so that the handheld user has to enter a number within the range. An example of this might be a grade or score that must lie between 0 and 100. The Advanced Field Property of Max & Min is used for this feature.</p> <p>Refer to Chapter 6, "Advanced Field Properties."</p>
Recommended	<p>Use Numeric fields for temperature readings, scores, or percentages, where the data to be entered is a numeric value. For percentages, you should specify in instructions to the user in the field name whether you want 10 percent to be entered as 10 or as 0.1.</p>

Continued

Continued

You can use Numeric fields to store the results of calculations on other numeric fields. Set the Read-Only Advanced Field Property to prevent the handheld user from modifying a calculated result. See Chapter 6, "Advanced Field Properties."

Not Recommended

Phone numbers and Social Security numbers are not ideal because you cannot enter hyphens (dashes) in the middle of a number. Use a Text field if you want to enter a number that is hyphenated, such as 123-45-6789

Works with AutoNavigate?

No

Currency Field

A Currency field (see Figure 5-3) requires the handheld user to enter a number with two decimal places. The following table provides an overview of the currency field.

Figure 5-3: A currency field on the handheld

Limits

A Currency field is a number with two decimal places. Commas are automatically added after every thousand.

The maximum length of a Currency field is nine digits, or +/- \$9,999,999.99

Alternatives

If you need to enter a currency value with more than nine digits, use a Numeric field instead.

Handheld Data Entry	In Field View, Pendragon Forms displays a custom numeric keypad on screen to enter the currency amount, or you can use Graffiti. In Record View, use Graffiti. The handheld's Numeric keyboard is not available in a Currency field.
Tips	If you perform a calculation using a Currency field, place the result in a Currency field also. If you put the result in a Text field or a Numeric field, the currency result will be displayed in cents, and you will need to convert the cents to dollars and cents.
Recommended	Use Currency fields to enter the cost of an item, or the amount of an expense, where the data to be entered is an amount in dollars and cents.
Works with AutoNavigate?	No

Yes or No Checkbox Field

A Yes or No Checkbox field (see Figure 5-4) actually gives the handheld user three possible choices to select: Yes, No, or leave the field blank. The following table illustrates the details of the Yes or No Checkbox field.

Field 10 of 18
Recommend to others?
Would you recommend our products to others?

Yes No

End Record View ◀ ▶

Figure 5-4: A Yes or No Checkbox field on the handheld

Limits	Yes is stored as the letter Y, and No is stored as the letter N. If the handheld user leaves the field blank, no response will be stored.
Alternatives	<p>If you need to have options such as Yes, No, and Maybe, use a Popup List field instead.</p> <p>Use a Popup List if you want to store the words Yes and No instead of Y and N.</p>
Handheld Data Entry	<p>In Field View, two check boxes are visible, one labeled Yes, the other labeled No. Tap an option to select it.</p> <p>In Record View, there is just one check box, which you can check to select Yes. See the following Tips for your options for recording a No response.</p>
Tips	If you enter data primarily in Record View, and you do not check a Yes/No Checkbox, the value in the field is blank. To record No, you need to check and then uncheck the box. You can also set a default value of N for No, so that No is recorded unless you check the box for Yes. To set a default value, see Chapter 6, "Advanced Field Properties."
Recommended	<p>Use Yes/No Checkboxes for fields such as checklist items.</p> <p>Yes/No Checkboxes are excellent for use with branching scripts, to allow you to see certain fields if you respond Yes, or to see other fields if you respond No. See Chapter 11, "Using Scripts."</p>
Works with AutoNavigate?	Yes

Option 1 of 5 Field

An Option 1 of 5 field (see Figure 5-5) displays five check boxes, labeled with the numbers 1 to 5; the handheld user can check a box to make a selection. The following table provides an overview of the Option 1 of 5 field.

Limits	There are five check boxes on screen, labeled 1, 2, 3, 4, and 5. The handheld user can check only one box.
--------	--

Field 7 of 18
Product Satisfaction
 On a scale of 1 to 5, 5 being the highest rating, how satisfied are you with our products?

1 2 3 4 5

End Record View ◀ ▶

Figure 5-5: An Option 1 of 5 field on the handheld

Alternatives	If you need to have options 1 to 10, use a Popup List.
Handheld Data Entry	Tap a check box to select that option.
Tips	<p>You can configure Option 1 of 5 to have fewer than five check boxes, or to display a different character from a number, by adding a character sequence in square brackets at the end of the field name. For example:</p> <p>What is the student's overall grade? [ABC]</p> <p>As shown in Figure 5-6, the character sequence displays on the handheld. However, one drawback to changing how the check boxes are labeled is that the data that is recorded and sent to the PC is still the numbers 1 to 5. So a selection of the letter A on the handheld will send the number 1 to the PC in this field. To make data entry as easy as possible for the handheld user, you may be willing to tolerate converting the number 1 to the letter A on the PC. If you cannot do conversion on the PC, use a Popup List or Lookup List instead, since these field types record what the user selects in the field.</p>
Recommended	<p>Use Option 1 of 5 fields for doing ratings. Remember to identify to the handheld user in the field name whether the 1 or the 5 is the best rating.</p> <p>Option 1 of 5 fields can be used with branching scripts, to enable you to jump to a particular section of a form depending on your selection.</p>
Works with AutoNavigate?	Yes

Field 4 of 4

Grade
What was the student's overall grade?

A B C D

End Record View ◀ ▶

Figure 5-6: An Option 1 of 5 field that has been modified to display letters instead of numbers.

Popup List Field

A Popup List field enables you to select an item from a list.

When you are designing a form, if you select Popup List as the field type, a box appears in the Form Designer window to enable you to enter the options that you want to appear in the Popup List.

As shown in Figure 5-7, each option in the list is entered on a separate line.

Form Designer

Form Name: Customer Satisfaction Survey

Field Name: Popup list

Field Type: Popup list

Field Label: Department
With which department did you interact?

Select one...

End Record View ◀ ▶

Field Options:

- Floor Sales
- Phone Sales
- Technical Support
- Shipping
- Other

Figure 5-7: A Popup List field in the Form Designer window

In Figure 5-8 the Popup List appears on the handheld. The following table outlines the specifications of Popup List fields.

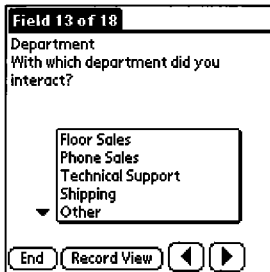


Figure 5-8: A Popup List in Field View on the handheld

Limits	<p>The maximum length of a Popup List field is 512 characters. Clicking Enter counts as two characters.</p> <p>Each item in the list cannot exceed 30 characters.</p> <p>When you freeze a form design, the entries in a Popup List are also frozen and cannot be changed unless you make a copy of the form and modify the copy. See Chapter 2, "Creating a Form," for information on copying forms.</p>
Alternatives	<p>If you need more than 512 characters for your list, use a Lookup List.</p> <p>If the options in your list are going to change on a regular basis, use a Lookup List instead of a Popup. Lookup Lists can be modified at any time without your having to copy the frozen form.</p>
Handheld Data Entry	<p>In Field View, tap the arrow labeled Select One to view the Popup List and make a selection.</p> <p>In Record View, tap the arrow in the field to view the list and make a selection.</p>
Tips	<p>If one value in a Popup List occurs more frequently than the other items, you can set that value as the default value in the field, to save time on data entry. See Chapter 6, "Advanced Field Properties."</p> <p>If you are entering records in Field View, you can use a branching script to gather more information if the handheld user selects certain options in a Popup List. For example, in a field with options Red, Blue, Green, and Other, if Other is selected, you can branch to a field that prompts for more detail. See Chapter 11, "Using Scripts."</p>

Continued

Continued

A Popup List can be made to work with a Lookup List in a Cascading Lookup relationship. This means that when you make a selection in the Popup List, your selection will determine which Lookup List is displayed in the next field. See the section on Cascading Lookup in this chapter.

Recommended

You can use Popup List fields that contain options that do not vary over time because the options are frozen with the form design.

You can use Popup List fields with branching scripts to jump to a particular section of a form depending on your selection.

Works with AutoNavigate? Yes

Multi-Selection List Field

In the Form Designer, a Multi-Selection List field looks similar to a Popup List field.

As shown in Figure 5-9, to create a Multi-Selection List, you need to enter the items in the list on separate lines.

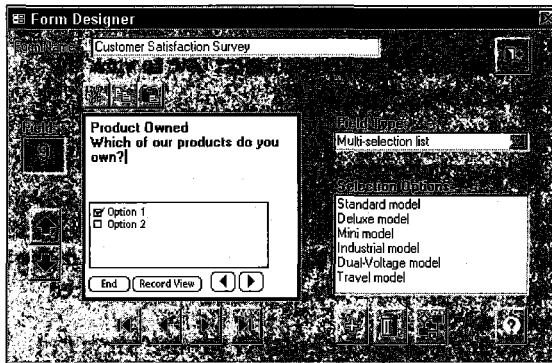


Figure 5-9: A Multi-Selection field in the Form Designer

Whereas a Popup List allows you to select only one item in the list, a Multi-Selection List (see Figure 5-10) allows you to choose one or more items by checking a box next to each item.

Figure 5-10: A Multi-Selection field on the handheld

The following table provides an overview of Multi-Selection fields.

Limits	<p>The maximum number of entries in a Multi-Selection List field is 32 items.</p> <p>Each item in the list cannot exceed 30 characters.</p> <p>Data coming back to the PC will contain each selected item, followed by a semicolon. Example:</p> <p>Red;Blue;Yellow;Lime Green;Purple</p> <p>When you freeze a form design, the entries in a Multi-Selection List are also frozen and cannot be changed unless you make a copy of the form and modify the copy.</p>
Alternatives	<p>If you need more than 32 items in your list, or if the items in your list will change regularly, use a Text field that references a Lookup List. See Chapter 6, "Advanced Field Properties."</p>
Handheld Data Entry	<p>Data can be entered in a Multi-Selection List field only in Field View. If you are in Record View and you tap in a Multi-Selection List field, the field will be displayed in Field View, and you can return to Record View after selecting items, by tapping the Record View button.</p> <p>To make selections in a Multi-Selection List field, check the box next to each item that you want to select.</p>
Tips	<p>If certain items in a Multi-Selection List occur more frequently than the other items, you can preselect the popular items as the default value in the field, to save time on data entry. See Chapter 6, "Advanced Field Properties."</p>

Continued

Continued

Recommended	You can use Multi-Selection fields that contain options that do not vary over time because the options are frozen with the form design.
Not Recommended For	Use with branching scripts. Because the handheld user can select any combination of items in the list, it would be difficult for a script to determine the action to perform in all possible cases.
Works with AutoNavigate?	No

Lookup List and Exclusive Lookup List Fields

Lookup Lists are similar to Popup Lists in that you can select an item from a list. However, Lookup Lists are more flexible than Popup Lists in two ways:

- ◆ A Lookup List can store many more items than a Popup List.
- ◆ You can change the items in a Lookup List after a form has been frozen.

With a Lookup List, if you do not want to choose one of the items in the list, you can type your own text in the field. An Exclusive Lookup List requires you to select from the list.

In order to have the flexibility to be modified after a form design has been frozen, Lookup Lists are created and stored separately from form designs. Using a Lookup List field is really a two-step process: creating the Lookup List, and then creating a Lookup List field on a form.



If you installed Pendragon Forms from the CD-ROM, open the Pendragon Forms Manager and select the Samples category. To see an example of a Lookup List field, distribute the form Project Tracker to your handheld. A Lookup List is ideal for a project list, because you can add items to the list at any time.

CREATING A LOOKUP LIST

To create a Lookup List, click the Lookups button in the Pendragon Forms Manager. The Lookup Editor window, as shown in Figure 5-11, is displayed.

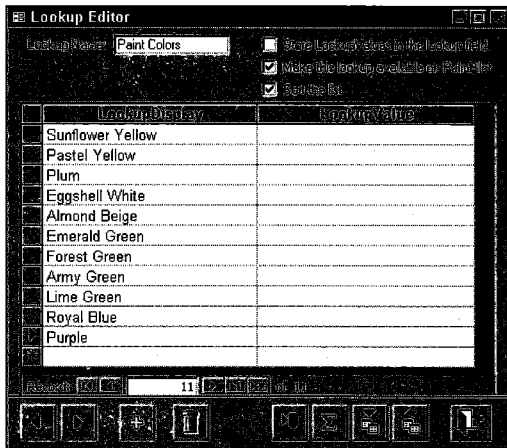


Figure 5-11: The Lookup Editor window

The left and right arrow buttons enable you to step through the Lookup Lists that have already been created. If you use the right arrow button to go past the last list, you will have a blank record to create a new list. Alternatively, click the + button to create a new Lookup List.

Type a name for the Lookup List in the Lookup Name field. Then click in the first cell of the LookupDisplay column.

If you want to create a list in which the value that is stored in the field is the same as the item that the handheld user sees in the list, then fill out only the LookupDisplay column, as illustrated in Figure 5-11.

The Lookup Editor window has three check boxes near the top of the window. The check boxes apply to the Lookup List currently being displayed. The check boxes are:

- ◆ **Make this Lookup available on PalmPilot.** This is the most important check box. If you want to send the Lookup List to the handheld, you must check this box.
- ◆ **Sort the List.** If you check this box, the list will appear in alphabetical order on the handheld.
- ◆ **Store Lookup Values in the Lookup Field.** It is possible to create a Lookup List that displays a list of items, but when the handheld user makes a selection, a different value is actually stored in the field. Figure 5-12 shows a Lookup List that will display the names of states, but when the handheld user makes a selection, what gets stored is the abbreviation for the selected state.

To accomplish this, the LookupValue column must be filled in, and the check box Store Lookup Values in the Lookup field must be checked.

Close the Lookup Editor window to save your Lookup List.

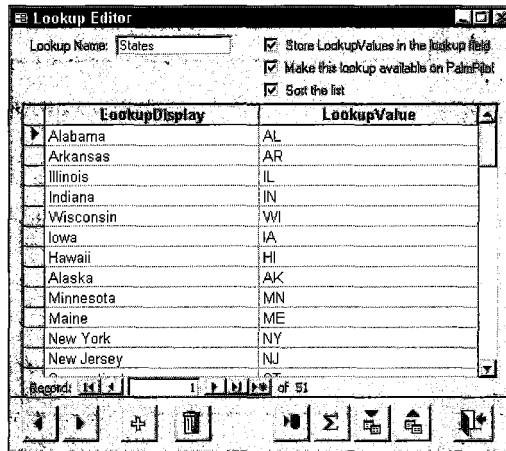


Figure 5-12: A Lookup List that stores Lookup Values



If you are using Lookup Values, each item in the LookupDisplay column must have a corresponding LookupValue. You can't leave any blank cells in the LookupValue column.

CREATING A LOOKUP LIST FIELD

Once a Lookup List has been created, you can create a form with a Lookup List or Exclusive Lookup List field.

In the Form Designer, as shown in Figure 5-13, select Lookup List or Exclusive Lookup List as the field type. In the Lookup List to Display field, select the Lookup List that you want to appear when the handheld user is in this field.

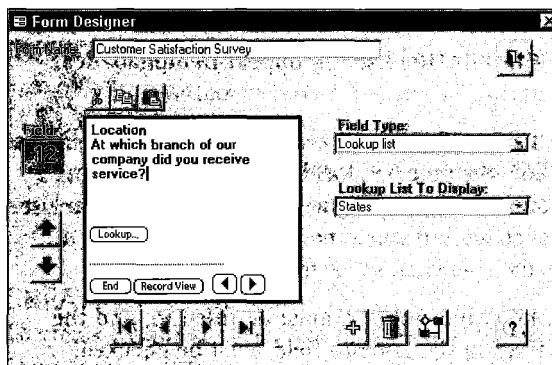


Figure 5-13: Creating a Lookup List field in the Form Designer window



When the form is on your handheld, if the Lookup List does not display, go back to the Lookup Editor window on the PC. Display the Lookup List and make sure that the check box *Make This Lookup Available on PalmPilot* is checked. If you are not using the *LookupValue* column, make sure that the *Store LookupValues* check box is not checked. Redistribute the form design and perform a *HotSync* data transfer.

Figure 5-14 shows a Lookup List field on the handheld. The following table provides an overview of Lookup List fields.

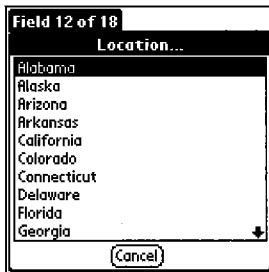


Figure 5-14: A Lookup List field on the handheld

Limits	<p>The maximum number of entries in a Lookup List is 1,000 items or 32 kilobytes.</p> <p>Each item in the list cannot exceed 50 characters. However, note that on the handheld screen, you can see only 36 characters across.</p> <p>The handheld user can select one item from the list.</p>
Alternatives	<p>If you need to select more than one item from a list, you can create several Lookup List fields on your form that all reference the same list.</p> <p>Alternatively, you can use a Text field that references a Lookup List. See Chapter 6, "Advanced Field Properties."</p>
Handheld Data Entry	<p>In Field View, tap the Lookup button to display the Lookup List. Tap an item in the list to select the item.</p>

Continued

Continued

	<p>In Record View, tap in the field to display the list.</p> <p>If a list is sorted in alphabetical order, you can use Graffiti to type a letter to jump to the first entry in the list with that letter. For example, writing the letter S in the Graffiti area highlights the first item in the list beginning with S.</p> <p>In a Lookup List field, if you do not want to select an item in the list, you can tap the Cancel button and then enter your own text. This works only in Field View. Note that if you enter your own text, the text does not get added to the list of items.</p>
Tips	<p>You can change the items in a Lookup List at any time even after a form has been frozen. On the PC, click the Lookups button to display the Lookup Editor window. Display the Lookup List and make the necessary changes. The updated Lookup List will be sent to the handheld during the next synchronization.</p> <p>You can also change which Lookup List is referenced in a Lookup List field on a form, even if the form design is frozen. In the Forms Manager, click the name of the form and then click the Edit button. Display the Lookup List field, and then reselect a different Lookup List to display. Save your changes and then redistribute the form design.</p> <p>A Lookup List can be used in a Cascading Lookup relationship with another Lookup List, or with a Popup List. In the cascading relationship, a selection in one field determines which Lookup List is displayed in the next field. See the "Cascading Lookup" topic in this chapter.</p> <p>Lookup Lists can be used to perform a lookup to another form or to copy fields from the reference form into the current form. See the "Lookup to Another Form" topic in this chapter.</p>
Recommended	<p>You can use Lookup List fields for fields that contain options that may vary over time, such as employee names or products sold.</p>
Not Recommended	<p>Short lists that do not change over time. A Lookup List uses the entire handheld screen to display the list. If you need a list with options such as A and B, and these options are not likely to change, then you should use a Popup List, which does not fill the entire screen to be displayed.</p>
Works with AutoNavigate?	Yes

DELETING LOOKUP LISTS

Deleting Lookup Lists requires care to ensure that the list is removed from the handheld.

To delete a Lookup List, click the Lookups button to view the Lookup Editor window. Click the right arrow button until you are viewing the Lookup List that you want to delete.

1. Un-check the box labeled Make This Lookup Available on PalmPilot. Then close the Lookup Editor window.
2. Perform a HotSync data transfer to remove the Lookup List from the handheld.

After you have performed a HotSync data transfer, you can delete the Lookup List from the desktop by displaying the Lookup List in the Lookup Editor window and then clicking the Delete button (Trash can icon).



In a multiuser environment, after you uncheck the box Make This Lookup Available on PalmPilot, you need to wait until all users have performed a HotSync data transfer before you can delete the Lookup List from the desktop.

Cascading Lookup

A *cascading relationship* can be established between a Popup List or Lookup List in one field, and a Lookup List in the next field. In the cascading relationship, a selection in the first field determines which Lookup List is displayed in the next field.



If you installed Pendragon Forms from the CD-ROM, open the Pendragon Forms Manager and select the Samples category. Distribute the form Calorie Counter to the handheld, to view a Cascading Lookup relationship between a Popup List and a Lookup List.

The left of Figure 5-15 shows the first field in a cascading relationship. In this case, the field is a Popup List showing food categories. Depending on the category selected in the first field, the Lookup List displayed in the next field is different. At the upper right, the Lookup List is related to fish, and at the lower right, the Lookup List is related to fruits.

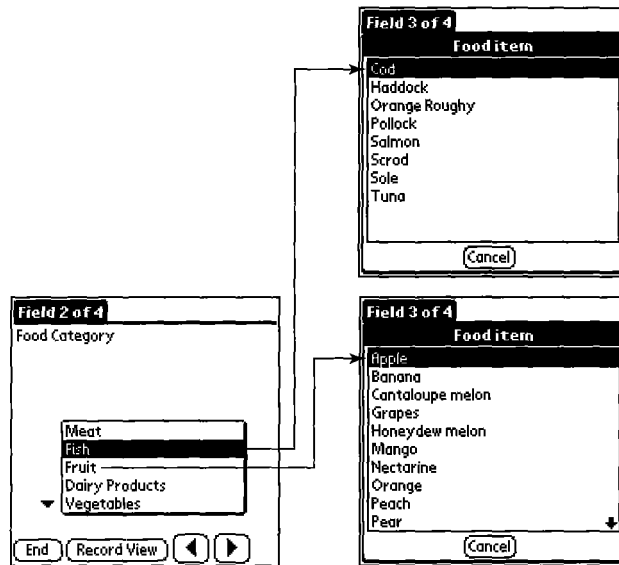


Figure 5-15: A selection in the first field in a Cascading Lookup relationship determines the Lookup List to be displayed in the next field.

To create a cascading relationship, the first field in the relationship can be a Pop-up List or a Lookup List containing the primary or category options.

For the next field on the form, create a Lookup List field. Instead of selecting a Lookup List to display, type a cascading keyword followed by the @ symbol. In Figure 5-16, the keyword is the word Food, because the categories are food-related. The keyword can actually be any word.

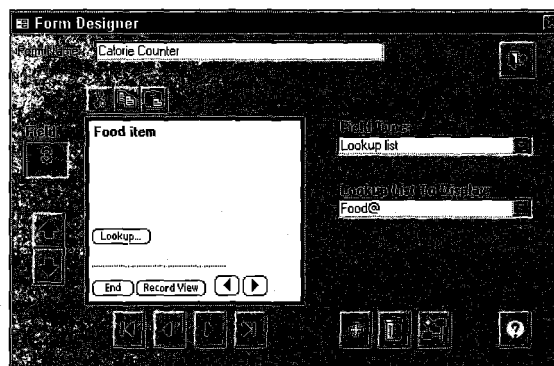


Figure 5-16: A Lookup List field with a keyword to create a cascading lookup

For each of the items in the primary list, create a Lookup List in the Lookup Editor. For example, there are five food categories in Figure 5-15, and so five separate Lookup Lists need to be created.

The name of each Lookup List is critical in order to create the cascading relationship.

- ◆ For each Lookup List name, type the cascading keyword followed by the name of an option in the primary list. For example, if the keyword is Food, and the primary options are Meat, Fish, Fruit, Dairy Products, and Vegetables, then the five Lookup Lists that need to be created are: FoodMeat, FoodFish, FoodFruit, FoodDairy Products, and FoodVegetables.
- ◆ The keyword name in the Lookup List must exactly match the spelling and case sensitivity of the keyword in the Lookup List field. Also, do not put a space between the keyword and the first word of the option name.

In the Lookup Editor for each of the cascading Lookup Lists, remember to check the box Make the Lookup Available on PalmPilot.

HOW A CASCADING LOOKUP WORKS

The Lookup List field in a cascading relationship has a keyword followed by the @ symbol, such as Food@.

When a selection is made in the field prior to the Lookup List field, the selection replaces the @ symbol to determine which Lookup List is displayed in the next field. For example, if Dairy Products is selected, then the Lookup List to be displayed in the next field is FoodDairy Products.



It is possible to nest cascading lookups. For example, one field can prompt you to select the type of hardware being installed, then the next field prompts you to select from a list of manufacturers for that hardware, and the third field prompts you to select from a list of models made by a given manufacturer. The danger in nesting cascading lookups is that the number of Lookup Lists expands at each cascading level. If there are five types of hardware, you need five Lookup Lists to store the manufacturers. If there are three manufacturers for each type of hardware, you need an additional fifteen Lookup Lists to store the models. This would mean twenty different Lookup Lists to maintain!

Lookup to Another Form

With a normal Lookup List, items can be added to the list only on the PC. If you need to be able to add items to a list on the handheld, you can use a separate form as the reference list and then perform a lookup to that form from another form. You can copy items from the reference form into the second form.

The advantages of using a lookup to another form over using a regular Lookup List are:

- ◆ You can add new items to the reference form at any time on the handheld.
- ◆ You can copy more than one field from the reference form to the other form.

The main disadvantage of performing a lookup to another form is that for each record on the form that is receiving data, you can store only the results of one lookup. This is because copying data from one form to another is achieved by having identical field names in the two forms. Once you perform a lookup and copy data from one form into the next, the fields will be filled. Performing a second lookup will only overwrite the same fields, not allow you to select a new item to copy elsewhere on your form.

Two steps are involved in creating a lookup to another form: creating the reference form, and creating the form that will perform the lookup to the reference form.



If you installed Pendragon Forms from the CD-ROM, open the Pendragon Forms Manager and select the Samples category. Distribute the forms Customer Account Info and Customer Contact Log to your handheld. The form Customer Account Info is a reference form, and the form Customer Contact Log is for recording individual visits with each customer. For each record of a customer visit, you can perform a lookup to the Customer Account Info form to select a customer.

CREATING A REFERENCE FORM

The reference form can contain any information. For inventory purposes, you can create a form containing item names, descriptions, and prices. For customer contact logs, you can create a form with customer information.

Figure 5-17 shows a form that has been created to store customer contact information.

Chopper Rescue	
Customer Name	Chopper Rescue
Account Number	C35412
Customer since	9/26/94
Account Standing	↓Excellent
Credit Limit	10,000
Contact Name	P. McOtter
Contact Phone	847-555-4567

End ⏪ ⏩ ⏴ ⏵

Figure 5-17: A reference form for customer contact information

Once the reference form has been created and frozen, you can populate the database table for the form by manually entering records on the PC, or by importing ASCII data into the form. Refer to Chapter 9, “Managing Data on the PC.”

If your reference information is contained in an external Access database, you can create a form that receives its data directly from your external database. Refer to Chapter 13, “Linking to an External Access Database.”



Remember that the default is that the handheld user can add new records to the reference form. If you prefer to populate the reference form from the PC, set the handheld access rights to No Additions on PalmPilot in the Form Properties window. See Chapter 3, “Entering Data on the Palm Organizer.”

CREATING A FORM TO LOOKUP TO THE REFERENCE FORM

Create a second form that will be used to perform the lookup to the reference form.

- ◆ For each field that you want to copy from the reference form, create a field on the second form with the identical field name and field type. The field names must have the same upper- and lowercase letters and spacing as the field names on the reference form.
- ◆ Also create a Lookup List field. Instead of selecting a Lookup List to display, type the name of the form which you want to look up. Figure 5-18 shows a Lookup List field that will be used to perform a lookup to a form called Customer Account Info.

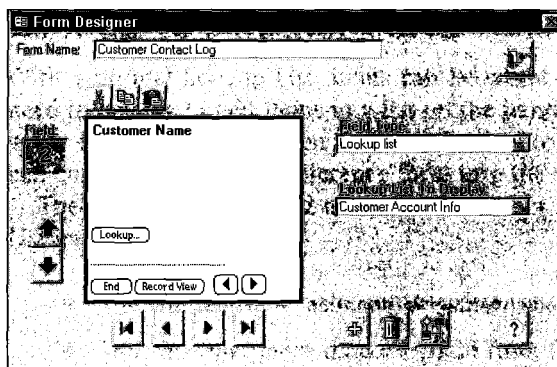


Figure 5-18: A Lookup List field being used to perform a lookup to another form



If the name of the Lookup List field identically matches the name of one of the fields on the reference form, then data will be copied into the Lookup List field. If the name of the Lookup List field does not match a field in the reference form, then the Lookup List field will remain empty after the lookup has been performed, but other fields on the form will contain data from the reference form. A Lookup List field can contain up to 50 characters.

HOW A LOOKUP TO ANOTHER FORM WORKS

Figure 5-19 shows a Customer Contact Log form, in which the first field on the form performs a lookup to a reference form. When the handheld user taps in the Customer Name field, a list of the records from the reference form is displayed, as shown in Figure 5-20.

11/4/99 9:30 am	
Visit Date & Time	11/4/99 9:30 am
Customer Name
Account Number
Contact Name
Contact Phone
Primary item discus
Other items discus
Follow-up required	<input type="checkbox"/>
Type of follow-up	↓
Proposed date of f	- No Date -
Comments on mee
<input type="button" value="End"/> <input type="button" value="←"/> <input type="button" value="→"/> <input type="button" value="⏪"/> <input type="button" value="⏩"/>	

Figure 5-19: A Customer Contact Log that performs a lookup to another form

Lookup	
11/4/99 9:30 am	
L Customer Account Info	
Alpha Quilts	
Bayview Lakes Homes	
Chopper Rescue	
Glory Brothers Inc.	
Janie Biscuits	
MavisWare	
Mini Things Inc.	
Rocket Science	
Sampson, Fenner & Pike	
<input type="button" value="Add"/> <input style="width: 100px;" type="text" value="Lookup:"/> <input type="button" value="Cancel"/>	

Figure 5-20: Instead of a Lookup List's being displayed, a list of records from the reference form is displayed.



The display key field from the reference form is displayed in the list. To view a second field in the list, tap the arrow in the upper-right corner of the handheld screen.

When you tap a record and then tap the Add button, all the fields that are identical on both the reference form and the current form are copied from the reference form to the current form. Figure 5-21 shows the lookup form with information copied in from the reference form.

11/4/99 9:30 am	
Visit Date & Time	11/4/99 9:30 am
Customer Name	Chopper Rescue
Account Number	135412
Contact Name	P. McOtter
Contact Phone	847-555-4567
Primary item discus	
Other items discus	
Follow-up required	<input type="checkbox"/>
Type of follow-up	↓
Proposed date of f	- No Date -
Comments on mee	
<input type="button" value="End"/> <input type="button" value="←"/> <input type="button" value="→"/> <input type="button" value="⏪"/> <input type="button" value="⏩"/>	

Figure 5-21: A Customer Contact Log form after copying data from the reference form

ADDING RECORDS TO THE REFERENCE FORM

If you perform a lookup to another form and the item that you want is not in the list of records in the reference form, you can add a new record to the reference form by tapping the Add button in the list. This will take you to the reference form to create a new record. When you end the record, you will return to the list of records, and you can select the new record to copy into the form that is doing the lookup.

Alternatively, you can add records to the reference form in the normal way—that is, by selecting the reference form from the Forms List on the handheld and tapping the New button.

Date & Time Field

A Date & Time field (see Figure 5-22) enables you to record the date and the time in a single field. The following table provides an overview of Date & Time fields.

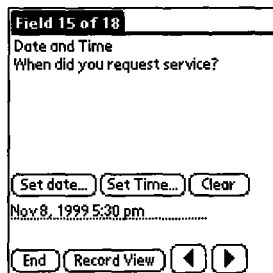


Figure 5-22: A Date & Time field on the handheld

Limits	The Palm organizer cannot accept dates before January 1, 1904, or after December 31, 2031. This is a limitation of the Palm operating system.
Alternatives	To enter a date outside the range available on the Palm calendar, you can use a Text field instead of a Date & Time field.
Handheld Data Entry	<p>In Field View, tap the Set Date button to select a date from an on-screen calendar. Tap the Set Time button to select a time. To cancel a selection, tap the Clear button.</p> <p>In Record View, tapping in a Date & Time field displays a calendar from which you can select a date. The current time is automatically entered. To select a different time, switch to Field view by tapping the name of a field in the left-hand column.</p>
Tips	<p>To set the default date to the current date and time, use an initialize: script.</p> <pre>Initialize: answer = now</pre> <p>See Chapter 11, "Using Scripts."</p> <p>To set the default date and time to a specific date — for example, 10/30/99 8:15 A.M. — set the Default Value of the field. See Chapter 6, "Advanced Field Properties."</p> <p>If you need to capture the creation date and time of a record, you can use Pendragon Forms' built-in TimeStamp field, which is viewable only on the PC.</p> <p>If you need to capture the date and time each time a record is modified, use an open: script. See Chapter 11.</p> <pre>open: answer = now</pre>

Recommended	You can use Date & Time fields for appointments where you need to record both a date and a time.
Not Recommended	<p>If you prefer to store the date separately from the time, use two fields instead — one Date Only field, and one Time field.</p> <p>Although you can perform calculations on Date & Time fields (see Chapter 11, "Using Scripts"), you cannot store the results of such a calculation in a Date & Time field. This is because a calculation such as End Date & Time minus Start Date & Time cannot be stored in an mmddyy or other date format. Use a Text field for storing calculated results.</p>
Works with AutoNavigate?	Yes

Date Only Field

A Date Only field (see Figure 5-23) records the date. The following table provides an overview of Date Only fields.

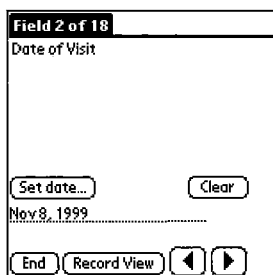


Figure 5-23: A Date Only field on the handheld

Limits	The Palm organizer cannot accept dates before January 1, 1904, or after December 31, 2031. This is a limitation of the Palm operating system.
Alternatives	To enter a date outside the range available on the Palm calendar, you can use a Text field instead of a Date & Time field.

Continued

Continued

Handheld Data Entry	<p>In Field View, tap the Set Date button to select a date from an on-screen calendar.</p> <p>In Record View, tap in a Date Only field to display a calendar and select a date.</p>
Tips	<p>To set the default date to the current date, use an <code>initialize: script</code>.</p> <p>See Chapter 11, "Using Scripts."</p> <p>To set the default date to a specific date — for example, 11/18/99 — set the Default Value of the field. See Chapter 6, "Advanced Field Properties."</p>
Recommended	You can use Data Only fields to enter birth dates or project milestone dates, where you need to record a date but not a time.
Not Recommended	Although you can perform calculations with dates (see Chapter 11, "Using Scripts"), you cannot store the results of such a calculation in a Date field. This is because a calculation such as Today's Date minus Birth Date cannot be stored in an mmdyy or other date format. Use a Text field for storing calculated results.
Works with AutoNavigate?	Yes

Time Field

A Time field (see Figure 5-24) records the time. The following table details the use of Time fields.

Figure 5-24: A Time field on the handheld

Limits	Records time in a clock format of hours and minutes, A.M. or P.M.
Alternatives	If you need to record days, use a Text field or a Numeric field.
Handheld Data Entry	In Field View, tap the Set Time button to select a time from an on-screen clock. To cancel entering a time, tap the Clear button. In Record View, tap in a Time field to display a clock and select a time.
Tips	To set the default time to a specific time — for example, 9:00 A.M. — set the Default Value of the field. It is necessary to add a date and a time to the Default Value, although only the time is recorded. See Chapter 6, "Advanced Field Properties." On the handheld, you can switch to displaying time in a 24-hour clock mode for applications such as flight logs. On the Palm Applications screen, tap the Preferences icon and then select the Formats option. In the Time field, choose HH:MM as the format for displaying time. This is a global setting that affects all Time fields on the handheld. Note that when data is uploaded to the PC, the data will be displayed in the format selected in the Windows Control Panel.
Recommended	You can use Time fields to enter meeting times or starting/ending times, where you need to record the time of day.
Not Recommended	Although you can perform calculations with times (see Chapter 11, "Using Scripts"), you cannot store the results of such a calculation in a Time field. This is because a calculation such as Completion Time minus Start Time cannot be stored in an A.M./P.M. format. Use a Text field for storing calculated results.
Works with AutoNavigate?	Yes

Time Checkbox

A Time Checkbox field (see Figure 5-25) looks like a Yes/No Checkbox, with options Yes and No. However, when you select Yes, the current date and time are recorded in the field. The following table provides an overview of Time Checkbox fields.

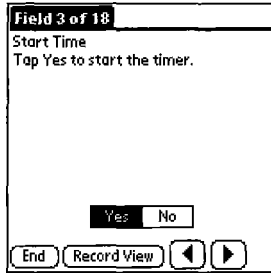


Figure 5-25: A Time Checkbox field on the handheld

Limits	<p>The current date and time (in hours, minutes, and seconds, A.M. or P.M.) is entered in the field when the handheld user selects Yes. The handheld user cannot see the recorded time.</p> <p>If the field is left blank, or if the handheld user selects No, a null value is placed in the field – no date or time.</p>
Alternatives	<p>If the handheld user needs to see the date and time, use a Date & Time field.</p>
Handheld Data Entry	<p>In Field View, two check boxes are visible, one labeled Yes, the other labeled No. Tap Yes to record the current date and time.</p> <p>In Record View, there is just one check box, which you can check to select Yes.</p>
Tips	<p>If you need to change the value in a Time Checkbox field, you can select No and then reselect Yes.</p>
Recommended	<p>You can use Time Checkbox fields to enter start and end times for recording the duration of a particular task.</p>
Not Recommended	<p>Although calculations can be performed on Time Checkbox fields, the result of such calculations should be placed in a Text field.</p>
Works with AutoNavigate?	Yes

Section Field

A Section field has two uses: the first is with Jump Popup fields, to enable you to jump to a section of a form. The second use of Section fields is for aesthetic reasons: to visually separate categories of fields on a form.

In addition to having a field name and a field type, a Section field also has a Section Name. The Section Name is used by Jump Popup fields. Figure 5-26 shows a Section field in the Form Designer window, with the Section Name filled in.

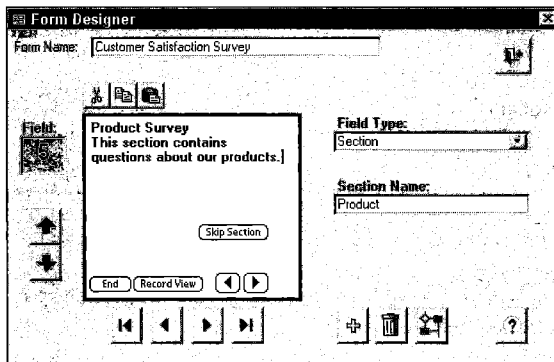


Figure 5-26: A Section field in the Form Designer window

The Jump Popup topic in this chapter will show you how Jump Popup fields are used to enable you to jump to a section field on a form.

Even if you are not using Jump Popup fields, you may want to consider using Section fields if you have a long form and you want to enter data in Record View.



If you installed Pendragon Forms from the CD-ROM, open the Pendragon Forms Manager and select the Samples category. Distribute the forms House Hunting A and House Hunting B to your handheld. Both forms contain the same questions, but House Hunting A has no Section fields, and House Hunting B has Section fields. View both forms in Record View to compare how Section fields visually aid the handheld user in identifying categories within the form.

Figure 5-27 shows a Section Field on the handheld. The following table provides an overview of Section fields.

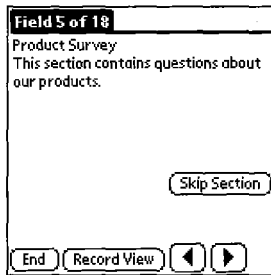


Figure 5-27: A Section Field on the handheld

Limits	<p>The Section Name cannot exceed 30 characters.</p> <p>A Section field is read-only: The handheld user cannot enter a response in the field.</p>
Handheld Data Entry	<p>In Field View, you can read instructions in the field name and tap the right arrow to move to the next field on the form.</p> <p>Tapping the Skip Section button jumps you to the next Section field on the form. If there are no more Section fields, you will end the record.</p> <p>In Record View, a Section field is the only field that is not in a two-column format. The field name is displayed in bold. This makes Section fields good markers for indicating separate categories on a form.</p>
Tips	<p>If you do not want to give the handheld user the ability to skip a section using the Skip Section button, you can hide this button. See Chapter 8, "Advanced Form Properties."</p>
Recommended	<p>Use a Section Field to delineate sections or topic categories within a form.</p>
Works with AutoNavigate?	No

Jump Popup Field

A Jump Popup field is a special type of Popup. It displays a list of sections of the form, and you can select to which section you want to jump. Section fields are required on the form in order to make the Jump Popup work.

In the Form Designer window, select Jump Popup as the field type. In the Popup Options field, type the names of the different sections, as shown in Figure 5-28.

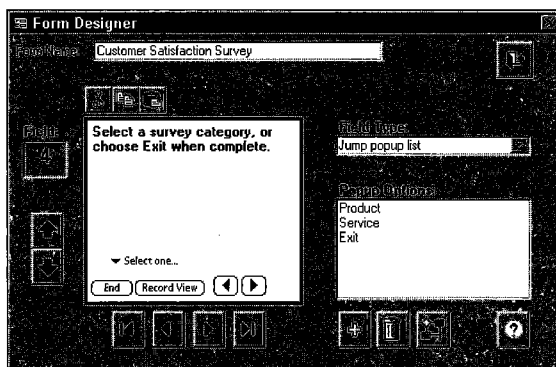


Figure 5-28: A Jump Popup field in the Form Designer window



If you installed Pendragon Forms from the CD-ROM, open the Pendragon Forms Manager and select the Samples category. Distribute the form House Hunting C to your handheld for an example of using a Jump Popup field to navigate to different sections of a form.

Figure 5-29 shows a Jump Popup Field on the handheld. The following table provides details on Jump Popup fields.

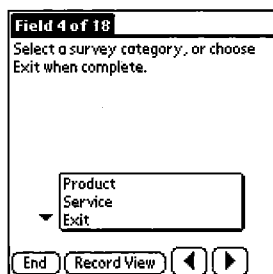


Figure 5-29: A Jump Popup Field on the handheld

Limits	<p>The maximum length of a Jump Popup list is 512 characters. Clicking Enter counts as two characters.</p> <p>Each item in a Jump Popup list must correspond to the Section Name of a Section field. The Section Name cannot exceed 30 characters.</p>
Handheld Data Entry	<p>In Field View, tap an item in the list to jump to the start of that section of the form.</p> <p>You can jump forward or backward to a section, so it is possible to position a Section field before a Jump Popup field. Typically the Section fields come after the Jump Popup field, because you want offer handheld users a choice of sections before they actually see the fields for a section.</p>
Tips	<p>After making a selection in a Jump Popup field and jumping to a section, you may want to return to the Jump Popup field to make another selection. If you are entering data in Field View only, you can add a branching script at the end of each section to return to the Jump Popup field. See Chapter 11, "Using Scripts."</p> <pre>exit: goto</pre> <p>You will need to have an Exit section or other similarly named section that allows you to stop branching back to the Jump Popup field but exit the form instead.</p>
Recommended	<p>Allowing the handheld user to fill in just the sections of the form that are necessary, leaving unused sections blank.</p>
Works with AutoNavigate?	No

Subform List Field

A Subform List field, also called a Subform field, enables you to create a "parent" and "child" relationship between two forms.

Typically the parent form contains information that does not vary much over time – customer information such as company name and account number, for example, or patient information such as name, address, and date of birth.

The child form or subform contains information about the parent record that varies constantly – individual customer or patient visits, for example.



If you installed Pendragon Forms from the CD-ROM, open the Pendragon Forms Manager and select the Samples category. Distribute the parent form Patient Info (Parent) and the subform Patient Visit Log (Subform) to your handheld.

CREATING A PARENT FORM AND A SUBFORM

Two separate forms need to be created – one will be the parent form, and the other will be the subform.

To create the link between the parent and the subform:

- ◆ On the parent form, create a field with field type Subform List. In the Popup Options section of the Subform field, as shown in Figure 5-30, type the name of the form that will serve as the subform. The form name cannot exceed 30 characters and must exactly match the name that you give the child form.

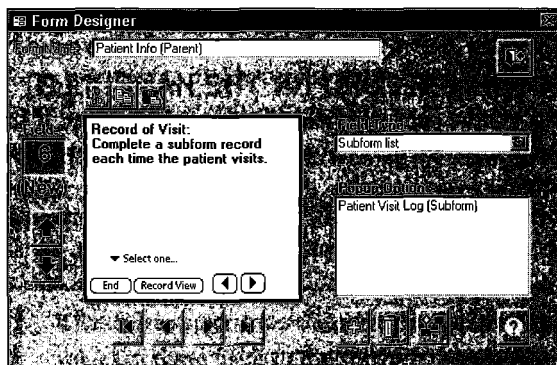


Figure 5-30: A Subform field in the Form Designer window

- ◆ At least one field on both the parent form and the subform has to match exactly. For example, you may want to put a Customer Name field on both parent form and subform. The field name (including upper- and lowercase letters and spaces), as well as the field type, must be identical in both the parent form and the subform.

The first field on each form is typically used as the identical field. Alternatively, one field from the first ten fields of the parent form must be identical to one field within the first ten fields of the subform.

- ◆ If you need to copy fields from the parent form to the subform, you can make up to the first ten fields on each form identical. For example, you may want to copy First Name, Last Name, Company, and Account Number from the parent record onto each subform record that you create.



A quick way to create a subform is to copy the parent form and then edit the copy to have a new form name. Delete all the fields that you do not want on the subform, and add all the fields that you do want on the subform.

- ◆ Select a Display Key for the subform. The Display Key is the field that will be used to display the subform records.
- ◆ By default, the Display Key is the first field on a form. However, if the parent form and subform have identical first fields, you may want to select a different field for the Display Key of the subform. For example, at the parent level you may want to view customer names, but at the child level you may want to view visit dates. The Display Key is an Advanced Field Property – see Chapter 6, “Advanced Field Properties.”

ENTERING A NEW SUBFORM RECORD ON THE HANDHELD

Subform records are created via the subform field of the parent record. If the parent record already exists on the handheld, you have to review the parent record in order to access the subform field.

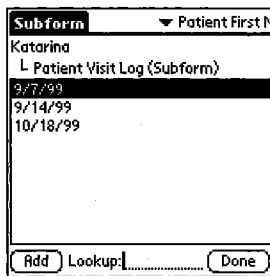
In Field View, the subform field looks like a Pop-up List field, as shown in Figure 5-31. In Record View, the subform field displays two small icons, as shown in Figure 5-32.

Figure 5-31: A Subform field in Field View on the handheld

Figure 5-32: A Subform field in Record View on the handheld

When you tap in the subform field on the parent form and select a subform in the list, a review screen displays a list of the subform records for that parent record. If there are no subform records, the list will be blank.

- ◆ To create a subform record, tap the Add button, shown in Figure 5-33.
- ◆ To review an existing subform record, tap the record.



The screenshot shows a handheld device interface. At the top, there is a title bar with 'Subform' on the left and a dropdown menu showing 'Patient First N'. Below the title bar, the text 'Katarina' is displayed. Underneath, there is a list item 'Patient Visit Log (Subform)'. Below the list, three dates are listed: '9/7/99', '9/14/99', and '10/18/99'. At the bottom of the screen, there are three buttons: 'Add', 'Lookup', and 'Done'.

Figure 5-33: When you tap in a Subform field, a list of the existing subform records is displayed.

- ◆ If a new subform record is being created, all of the fields from the parent form that match fields on the subform will be copied from the parent record to the new subform record.
- ◆ Pendragon Forms uses the matching fields between parent form and subform to determine which subform records belong to which parent record.



If fields are copied from the parent to the subform and then you change the values of these fields in the subform record, the record will not be listed in the parent form review screen when you review existing subform records. This is because the program displays only records that match on all identical field names. To protect the copied fields, you can make them read-only on the subform.

Figure 5-34 shows a new subform record with fields copied from the parent form. The handheld user can then complete the subform record.

Figure 5-34: A new subform record

When you tap the End button to end the subform record, you will be returned to the list of subform records. Tap the Done button to return to the parent record, and then tap the End button to exit the parent record.

The following table provides an overview of Subform fields.

Limits	On the parent form, the name of the referenced subform cannot exceed 30 characters.
	You can list more than one subform in the Popup Options field of the parent – the entire list cannot exceed 512 characters.
Alternatives	An alternative to using a parent form and a subform is to use two forms and perform a lookup to another form.
Handheld Data Entry	Always enter data starting from the parent form.
Works with AutoNavigate?	No

WORKING WITH EXISTING SUBFORM RECORDS

When you tap in the subform field of a parent form, a list of existing subform records is displayed, as shown in Figure 5-35.

The Display Key field is used to display the records. If the Display Key is not sufficient to identify which record you want to review, you can tap the arrow in the upper-right corner of the screen, and select a second field on the subform to review, as shown in Figure 5-36.

Date of Visit
9/7/99
9/14/99
10/18/99
10/25/99
11/8/99

Figure 5-35: The Display Key field is used to display a list of existing subform records.

On Medication	
9/7/99	Y
9/14/99	N
10/18/99	N
10/25/99	Y
11/8/99	Y

Figure 5-36: You can select to view a second field in addition to the display key.

Why Must Subform Records Be Created from the Parent Form?

On the handheld, the parent form and the child or subform are displayed in the Forms List as separate forms.

To create a new subform record, you cannot select the subform in the Forms List, and choose to create a new record the way that you would for a regular form. Instead, you must review the parent record and then tap in the subform field on the parent. This is because data has to be copied from the parent to the subform record.

The data that is copied from parent form to subform is at least one field and can be up to the first ten fields of the parent. The program uses these matching fields to determine which subform records belong to which parent record.

If you create a subform record independently of the parent form, there is a risk that you may not enter data exactly the same as the parent form in the fields that are used to link the two. If you make an error, the subform record will not be associated with the parent record, and you will not be able to review the record from within the parent form.

To prevent accidental data entry directly into the subform, you can choose to display the subform differently in the Forms List on the handheld, or to hide the subform from the Forms List. Refer to Chapter 8, "Advanced Form Properties."

Another feature that can be used to find a record is the Lookup feature of the subform review screen. If you enter a character in the Lookup field on the subform review screen, the first record beginning with that character will be highlighted. The search is performed on the Display Key field.



If you want to change the Display Key of a subform, select a new Display Key in the Advanced Field Properties screen on the PC, and then redistribute the form to the handheld. See Chapter 6, "Advanced Field Properties."

Once you have found the subform record that you want to review, tap the record to view its contents. You can then review or modify the record as needed.

As shown in Figure 5-37, a subform record includes a small icon of two connected arrows in the upper-right corner of the screen. When you tap this icon, as shown in Figure 5-38, you can view the name of the parent form and the display key and primary key of the parent record. This information can give you a bearing of which parent record you are working with, without your having to leave the subform record.

10/18/99

Patient First Name Katarina
 Patient Last Name Henries
 Social Security Nu 345-67-8910
 Date of Visit: 10/18/99
 Patient Symptoms
 On Medication?
 Name of Medicatio
 Physical exam per
 Comments on phys
 Blood Sample requi
 Urine sample requi
 End

Figure 5-37: Reviewing a subform record

10/18/99

Parent Form Info
 Katarina
 Name of Medicatio
 Physical exam per
 Comments on phys
 Blood Sample requi
 Urine sample requi
 End

Figure 5-38: Viewing parent form details from a subform record

PARENT AND SUBFORM DATA ON THE PC

Parent forms and subforms only appear to be connected on the handheld. In reality, the two are separate forms. This means that in the database on the PC, there are two separate, unconnected database tables for storing the information from the parent form and from the subform.

One of the reasons why data needs to be copied from parent record to each new subform record is that when the data comes back to the PC, all subform records are stored in the same database table. In order to identify which subform record belongs to which parent, the subform needs to have at least one identifiable field from the parent record, such as a customer name.

Using Multiple Subforms and Nesting Subforms

You can choose to have one parent form with several subforms.

On the parent form, you can create a separate subform field for each subform. Alternatively, because a subform field contains a pop-up list of options, you can choose to list several subforms in the same subform field on the parent form.

You can also nest subforms, as long as you do not reference the same form name twice. Nesting subforms means that from a parent form you can jump to a subform, and then in that subform you can have a subform field that allows you jump into another subform.

REMOVING PARENT FORM AND SUBFORM DATA FROM THE HANDHELD

On the Palm organizer, parent forms and subforms appear to be linked – in reality, the program on the handheld considers them as independent forms. This means that if you have a Completion Checkbox on a parent form, for example, and you mark a parent record for removal from the handheld, the subform records associated with that parent will not automatically be removed from the handheld.

The best way to handle the removal of subform records is to set a limit to the length of time that subform records remain on the handheld. Because no correlation exists between parent and subform records, it is possible that subform records may be removed before their corresponding parent record, or that subform records will remain on the handheld after the corresponding parent records have been removed from the handheld. However, in most cases, an expiration time limit for subform records can be chosen to make a good compromise between the two cases. For example, if parent records are removed after an average of 7 days, you might find that a time limit of 10 days works best for the subforms.

Refer to Chapter 2, “Creating a Form,” for information on Data Persistence options for specifying how long records remain on the handheld.

Single Subform Field

A Single Subform is a special type of subform, one in which you can only create one subform record per parent form.

The primary purpose of a single subform is to provide a way to lengthen a form. Each form design that you create is limited to 250 fields, because of the limit to the number of database columns that a Microsoft Access database table can contain. If you have a form that needs to exceed 250 fields, you can allocate sections of the form to a single subform. Each single subform is a separate form design and can hold up to 250 fields also.

Another use for single subforms is to break a form into manageable units on the handheld. For example, an inspection may consist of several separate tasks. A parent form can be used to display all of the tasks on one form, and tapping in a field for an individual task jumps you to a single subform that displays the detail for that task.

To create a single subform, follow the instructions for creating a parent and subform, with one exception: Instead of using a Subform List as the field type on the parent form, select Single Subform List as the field type, and select or type the name of the form to be used as the single subform.

As in a regular parent and subform, with a parent and single subform there must be at least one field from the first ten fields on the parent form that exactly matches one of the first ten fields on the single subform.



If you installed Pendragon Forms from the CD-ROM, open the Pendragon Forms Manager and select the Samples category. For an example of a parent form and single subforms, distribute the following three forms to your handheld: Insurance Review, Life Info, and Car Info. Insurance Review is a parent form, and Life Info and Car Info are single subforms.



To make the parent form display in Record View format when creating new records, set the Advanced Form Property of Default to Record View. (See Chapter 8, "Advanced Form Properties.")

HOW A SINGLE SUBFORM WORKS

As shown in Figure 5-39, the parent form contains fields for the single subforms. Before a single subform is filled in, the icon in the field appears as a blank page with an upturned corner.

Tap in a single subform field to display the subform, as shown in Figure 5-40.



Remember that as with regular subforms, with single subforms there are separate database tables on the PC for the parent form and for each single subform. If all the fields on a parent form and its single subforms can actually fit on a single form, it may be overkill in a single-user environment to maintain separate database tables for sections of the form. In a multiuser environment it may be advantageous to make a form as easy to use as possible on the handheld, and then to manage joining information from different database tables on the PC.

J. Louis

Customer Name J. Louis
 Date of Visit 11/11/99
 Address 14 Maracas Bay Rd
 Phone 644-555-1311
 Life Insurance
 Car Insurance

End ⏪ ⏩ ⏴ ⏵

Figure 5-39: A parent form with single subform fields

Jane


Customer Name J. Louis
 Date of Visit 11/11/99
 Current Life policy
 Need Life Policy?
 Need Term Policy?
 Date of Birth 11/12/54
 Smoke?
 Medical?
 Plans discussed

End ⏪ ⏩ ⏴ ⏵

Figure 5-40: Selecting a single subform field displays the single subform.

After you have ended a single subform record and you return to the parent form, the icon in the field will display a page with lines, as shown in Figure 5-41. This indicates that the single subform has been filled in.

J. Louis

Customer Name J. Louis
 Date of Visit 11/11/99
 Address 14 Maracas Bay Rd
 Phone 644-555-1311
 Life Insurance 
 Car Insurance

End ⏪ ⏩ ⏴ ⏵

Figure 5-41: A parent form with a filled-in single subform

The following table provides an overview of Single Subform fields.

Limits`	On the parent form, the name of the referenced single subform cannot exceed 30 characters.
Handheld Data Entry	Always enter data starting from the parent form.
Works with AutoNavigate?	No

Signature Field

A Signature field enables you to capture a signature on the handheld.

When data is uploaded to the PC, each signature is stored as a Long Binary Object in the Forms3.mdb database (or Forms32k.mdb if you are using Microsoft Access 2000).

- ◆ In the Edit/View window on the PC, if you double-click in a Long Binary Object cell, Windows Paint will open to enable you to view the image. You can then save the image as a bitmap (.bmp) file if necessary.
- ◆ If you are using Microsoft Access to create reports, note that Pendragon Forms has a Report Generator that converts the bitmap to an image that you can see on your report. Refer to Chapter 9, “Managing Data on the PC.”

Figure 5-42 shows a Signature field in the Form Designer window. When you select Signature as the field type, a Legal Terms box appears. You can type some information in this box to inform the person whose signature is being captured of the terms to which they are signing their agreement.

If you do not want to enter anything in the Legal Terms box, you can leave it blank.

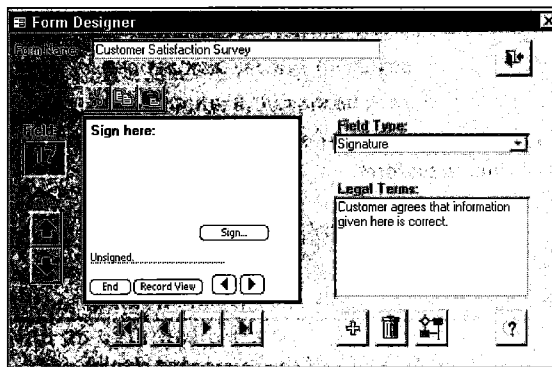


Figure 5-42: A Signature field in the Form Designer window



You should consult your legal advisor about the legality of displaying agreement terms and capturing a signature on a handheld device. There may be additional issues involved, such as giving a customer a printed copy of the signed document.



If you installed Pendragon Forms from the CD-ROM, open the Pendragon Forms Manager and select the Samples category. For an example of a form with a Signature field, distribute the form Package Delivery to your handheld.

ENTERING A SIGNATURE ON THE HANDHELD

If you tap in a signature field, the screen displays the message “Signed” if a signature has been entered, or “Unsigned” if the field is blank, as shown in Figure 5-43.

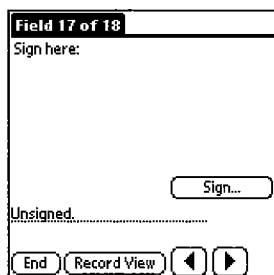


Figure 5-43: A Signature field on the handheld

To enter a signature, tap the Sign button. If the Legal Terms box has been filled in, you will see a screen such as that shown in Figure 5-44, displaying the terms to which the signatory is agreeing. Tap the Agree button to signify agreement, and then a signature screen will be displayed, as shown in Figure 5-45.

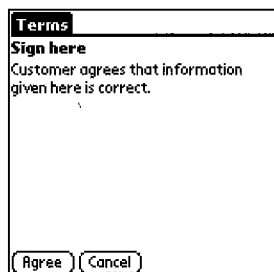


Figure 5-44: If there are agreement terms, these terms will be displayed before you sign in a Signature field.

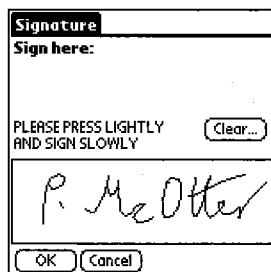


Figure 5-45: Capturing a signature on the handheld

Use the stylus to sign in the box that appears on the handheld screen. If you are not satisfied with the signature, tap the Clear button to erase it and try again. If you are satisfied, tap OK. If you are not ready to sign at this time, tap Cancel.

The following table provides an overview of the Signature field.

Limits	The text in the Legal Terms section of a signature can be up to 2,000 characters.
Handheld Data Entry	Tap in the Signature field to capture a signature. See the preceding topic.
Tips	You can have more than one Signature field on a form.
Works with AutoNavigate?	No

Button Field

A Button field displays a button on screen, and when you tap the button, a script is run to perform an action.

- ◆ Button fields can be used only in conjunction with `click:` scripts. See Chapter 11, “Using Scripts.”
- ◆ To create the label on a button, enter a word as the Default value in the Button field. See Chapter 6, “Advanced Field Properties.”

Actions that can be performed using buttons are:

- ◆ Performing a Lookup to the Palm Address Book, to copy information to your form
- ◆ Performing calculations that run only when the button is tapped
- ◆ Printing a record
- ◆ Attaching a record to an e-mail
- ◆ Transmitting a record to a Web site using Palm VII technology

Examples of the scripts used for these actions are covered in Chapter 11, “Using Scripts,” and Chapter 16, “Working with the Palm VII.”



If you installed Pendragon Forms from the CD-ROM, open the Pendragon Forms Manager and select the Samples category. For an example of a Button field, distribute the form Address Book Lookup to your handheld.

Figure 5-46 shows a form that contains a Button field. In this example, when you tap the Button field, an Address Book list is displayed, and you can select a person from the list. Making a selection copies data from the Address Book into the form, as shown in Figure 5-47.

Figure 5-46: A form with a Button field

Figure 5-47: Tapping the button performs an action, in this case enabling you to copy Address Book information into the form.

The following table provides an overview of the Button field.

Limits	The label on a button is limited to 14 characters. A label is created by entering text into the Default Value of a Button field. See Chapter 6, "Advanced Field Properties."
Handheld Data Entry	Tap the button to execute the script associated with the button.
Tips	In the <code>click:</code> script for the Button field, you can set the label on the button, so that after the handheld user taps the button, a different label is displayed. You might want to change the label if you want the user to know that the button has been tapped. Refer to Chapter 11, "Using Scripts."

Continued

Continued

Recommended	Use a Button field to create scripts that the handheld user can control when the script is run.
Works with AutoNavigate?	No

READ-ONLY TEXT FIELD

Read-Only Text fields are primarily included for backward compatibility with earlier versions of Pendragon Forms.

Instead of using Read-Only Text fields on your form, you can now set a Read-Only property on a field-by-field basis. See Chapter 6, "Advanced Field Properties."

Read-Only Text fields were originally designed to give the handheld user information or instructions. The information that is displayed can fit only in the field name, however, which is about 127 characters.

COMPLETION CHECKBOX FIELD

A Completion Checkbox is a special type of Yes/No field.

When you select Yes in a Completion Checkbox field, the record is marked for removal from the handheld during the next HotSync data transfer.

For records to be removed when a Completion Checkbox is checked, the data persistence option of Keep Incomplete Records on Handheld has to be selected in the Form Properties window, as shown in Figure 5-48. To access the Form Properties window, click the name of the form in the Forms Manager and then click the Properties button.

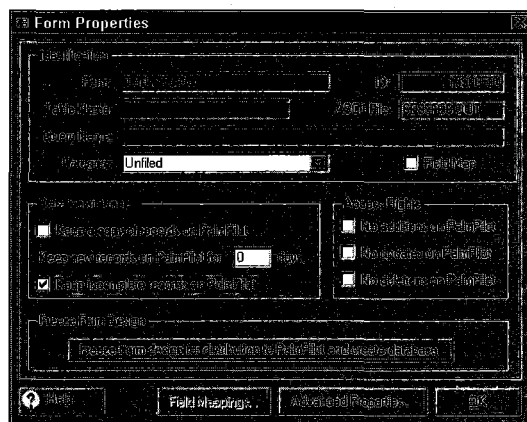


Figure 5-48: The Form Properties window



If you installed Pendragon Forms from the CD-ROM, open the Pendragon Forms Manager and select the Samples category. For an example of a form with a Completion Checkbox, distribute the form Tasks To Do to your handheld. Notice that records marked Yes in the Completion Checkbox field are removed from the handheld after a HotSync data transfer, and records with the Completion Checkbox field marked as No or left blank remain on the handheld.

Figure 5-49 shows a Completion Checkbox field in Field View on the handheld. The following table provides an overview of the Completion field.

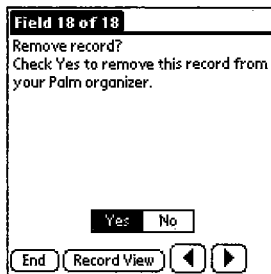


Figure 5-49: A Completion field on the handheld

Limits	Select Yes to flag a record for removal from the handheld. The letter Y is recorded in the field for Yes, N for No, and the field remains blank if neither Yes nor No has been selected.
Alternatives	After a HotSync data transfer, the record will be removed from the handheld if the field has been marked as Y for Yes. If you did not mean to remove the record from the handheld, enter N in the field on the PC, and the record will be sent back to the handheld after another HotSync data transfer.
Handheld Data Entry	In Field View, tap Yes to remove the record from the handheld during the next synchronization. To keep the record on the handheld, leave the field blank, or select No.

Continued

Continued

	In Record View, check the check box to mark the record for removal, or leave the check box blank to keep the record on the handheld.
Tips	You can have only one Completion Checkbox on a form. If there is more than one, then checking any Completion Checkbox will flag the record for removal from the handheld.
Recommended	Use a Completion field for work order forms in which you want to keep a record on the handheld until the work is complete.
Not Recommended	Use in recording general Yes and No responses. Use a Yes/No field instead.
Works with AutoNavigate?	Yes

Summary

This chapter showed you the 21 field types that can be used when designing a form, and the limitations of the various field types. You also saw how data is entered on the handheld depending on the field type.

Chapter 6

Advanced Field Properties

IN THIS CHAPTER

- ◆ Naming database columns
- ◆ Setting up default values in fields
- ◆ Selecting a Display Key
- ◆ Making fields hidden
- ◆ Requiring the handheld user to fill in a field
- ◆ Using Primary Keys
- ◆ Setting a maximum length of Text fields and Numeric ranges

FOR EACH FIELD ON a form, you can set Advanced Field Properties to gain additional control over the field.

The types of control that you have over individual fields include:

- ◆ Setting a default value in a field, to minimize the need for data entry
- ◆ Making a field read-only, so that the handheld user cannot modify a calculated result
- ◆ Hiding a field, and then using a script to show the field if the handheld user needs to fill in that field
- ◆ Making a field required, so that the handheld user does not omit entering data in the field
- ◆ Choosing whether a field prints when a record is printed from the handheld

Some Advanced Field Properties apply to every field type. Other Advanced Field Properties apply to a specific field type, such as:

- ◆ In a Text field, you can set the field to accept as few as one character, or as many as 2,000 characters.
- ◆ In a Numeric field, you can set the field to accept only numbers within a specified range, or whole numbers only.



Most Advanced Field Properties can be modified after a form is frozen. Save changes to the form design, and then redistribute the form in order for the modified Advanced Field Properties to take effect on the handheld.

The Advanced Field Properties window is accessed from the Form Designer window when you create or edit a form. In the Form Designer window, display the field whose Advanced Field Properties you want to set. Then click the Advanced Field Properties button to display the Advanced Field Properties window for the field (see Figure 6-1).

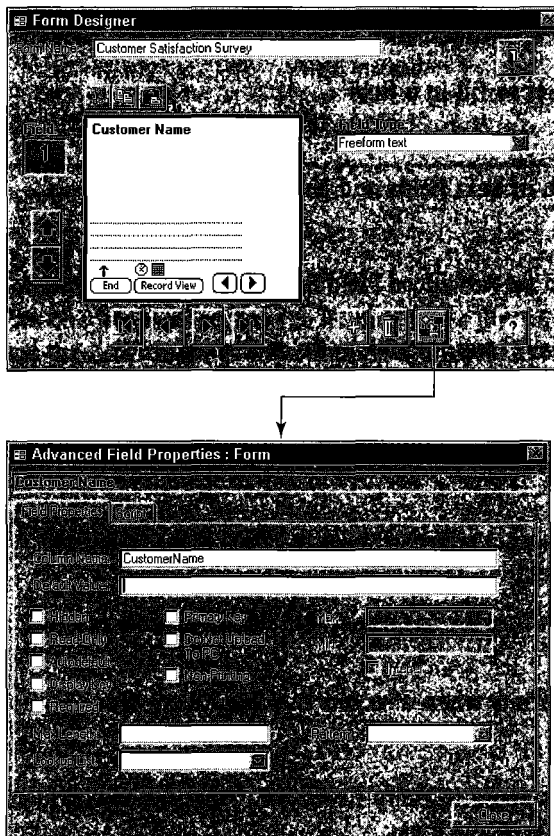


Figure 6-1: Clicking the Advanced Field Properties button in the Form Designer window displays the Advanced Field Properties window.

If an Advanced Field Property is grayed out, it does not apply or cannot be modified for a given field.

Each Advanced Field Property is described in detail in this chapter.

Column Name

When you freeze a form design, a database table is created for storing the records associated with the form design. For each field that you have on your form, there is a column in the database table.

The Column Name field in the Advanced Field Properties window records the name of the database column for each field.

Before you freeze a form, the Column Name is blank. If you keep this default, then when you freeze the form design, Pendragon Forms will generate a column name based on the name of the field.

- ◆ The Column Name is the same as the field name, up to the first 64 characters. If the field name contains spaces and nonalphanumeric characters, these will not be included in the Column Name.
- ◆ If two fields have the same name up to the first 64 characters, the Column Name will be modified so that each database column name is unique. This is a Microsoft Access requirement.

If you prefer to use your own names for the database columns, then before you freeze the form, type a name in the Column Name field for each field on your form. (Database design principles recommend that you not use spaces in a Column Name.) When you freeze the form, your database column names will be used instead of the auto-generated Column Names.

If there are duplicate column names, the program will not allow you to freeze the form design, and you will need to edit the form so that each Column Name is unique.



If you need to troubleshoot error messages that occur while you are trying to freeze a form design, see Appendix A, "Troubleshooting Tips."



Once you freeze a form, you cannot change the Column Names.

If you are linking to an external Access database or ODBC database, when you create a form based on the existing database table, Pendragon Forms automatically assigns the same Column Names as are used in the existing database table.



For information on linking to an external Access database, see Chapter 13.

For linking to an ODBC database, see Chapter 14.

Default Value

A Default Value is a value that will be entered into a field unless the handheld user chooses to enter a different value.

Default Values have three main uses:

- ◆ To speed up data entry in fields where the response is almost always the same. For example, if there is a Popup List field with options New and Used, and nine times out of ten the response is New, then you can make New the default value.

What Happens to Column Names When I Copy a Form?

When you copy a form, the Column Names of the original form are retained. This enables you to easily import data from the original form into the copy, if necessary. Refer to Chapter 9, "Managing Data on the PC," for information on importing data from one form design into another.

Because a copied form is not frozen, you can change the field names of the fields. However, this does not automatically change the Column Names. If you are not importing data from the original form, and you need the Column Names of the copied form to reflect the new field names, then delete the Column Name of each field. Then when you freeze the form, the program will auto-generate Column Names based on the new field names.

On occasion, a duplicate Column Name situation may arise when you are working with a copied form. If a field name is changed on the copy, the underlying Column Name is not changed. If a new field is added to the form with the same name as the original field, then when you try to freeze the form there will be two Column Names that are identical, even though the two field names are different. Pendragon Forms will not allow you to freeze the form, and the solution is either to erase all Column Names or to change one of the duplicates, and then freeze the form.

- ◆ To overcome ambiguity if the handheld user does not make a selection in a field. For example, in a Yes/No Checkbox, if the user does not make a selection, a blank value is stored in the field. By setting a default value of No, you can eliminate the ambiguity of a blank response, making the value always No unless the user selects Yes.
- ◆ In a Button field, you can assign a label to the on-screen button by entering a Default Value.



Because a Default Value places an entry into a field, the handheld user must read the response in order to identify whether to change the value in the field. If the handheld user is operating under fatigue or under distracting conditions, there is a risk that the user might accidentally accept the Default Value and move on to the next field without realizing that a change should have been made in the field. If you think your users might work under such conditions, it may be better not to use Default Values.

To enter a default value, type in the Default Value field of the Advanced Properties window. Table 6-1 illustrates the type of information that you can enter depending on the field type.

When you exit the Advanced Field Properties screen, a check will be performed to ensure that the default value that you entered is valid.

TABLE 6-1 VALID DEFAULT VALUES

Field Type	Valid Default Value Format
Text	Any characters, up to 255.
Numeric	Any number up to 15 digits. Can include minus sign or exponent. Note that if you enter a default of 6.4, for example, and you also have the Advanced Field Property of Integer set, the 6.4 value will display on the handheld, but if the user makes changes he/she can enter only an integer.
Currency	A number with two decimal places, up to 9 digits (including the decimal places).

Continued

TABLE 6-1 VALID DEFAULT VALUES (Continued)

Field Type	Valid Default Value Format
Yes or No CheckBox	Type Y for a default of Yes. Type N for a default of No.
Popup List	Type the name of one of the options in the Popup List. The name must exactly match that of an item in the list, including upper- and lowercase letters.
Lookup List	Type the name of an item in the Lookup List, or type alternative text up to 50 characters.
Exclusive Lookup List	Type the name of an item in the List.
Option 1 of 5	Type a number from 1 to 5.
Date Only; Date & Time	You can set a default to a specific date, not the current date. (See sidebar for information on defaulting to today's date.) Type a date in the Short Date Format used in Windows. For example, if your Short Date Format uses a four-digit year, and you want to set a default of September 15, 1999, enter 09/15/1999. In a Date & Time field, include a time after the date, in the Windows Short Date/Time format, for example: 09/15/1999 10:30 A.M.
Time	You can set a default to a specific time, not the current time. (See the sidebar for information on defaulting to the current time.) To make a Time field default to a specific time, you need to enter a date as well. For example, if you want a default time of 9:00 A.M., you can enter the following in the Default Value field: 7/31/1999 9:00 A.M.
Multi-Selection List	Type the names of the items in the list, separated by semicolons. For example: Red;Blue;Purple
Button	The Default Value in a Button field is the label on the button. Type up to 14 characters.

Setting a Default to the Current Date and Time

The Default Value field is used to default to a specific date and time, not the current date and time.

If you want to default to the current date and time, you can use a script. Refer to Chapter 11, "Using Scripts."

An `initialize` script can be used to set the current date and/or time when the record is created. The script is as follows:

```
initialize:  
answer = now
```

Hidden

A Hidden field is one that the handheld user ordinarily does not see in Field View or Record View.

Typical uses for Hidden fields are as follows:

- ◆ For use with scripts. For example, fields pertaining to a certain category of questions can be made hidden, and then if the handheld user selects an option to fill out that category, a script runs to show the hidden fields.
- ◆ To store intermediate calculated results that the handheld user does not need to see
- ◆ To record the last time the handheld user modified a record

To make a field hidden, check the Hidden check box in the Advanced Field Properties window for the field.

Keeping Track of the Modification Time of a Record

Pendragon Forms stores the creation date and time of a record in a `TimeStamp` field. In some forms, however, you may also need to know when the handheld user has made a change to a record. You may choose to record this information in a Hidden field, so that the handheld user cannot alter the modification date time.

Continued)

Keeping Track of the Modification Time of a Record (Continued)

Recording a modification date and time requires a script. Refer to Chapter 11, "Using Scripts," for more information on scripts.

Create a Text field and set the Advanced Field Property of Hidden. Then create a calculate: script that runs every time a field has been changed. The following calculate: script records the date and time of the last modification of the record:

```
calculate:  
answer = now
```

If you prefer the handheld user to be able to see but not change the last modification date, use the Advanced Field Property of Read-Only instead of Hidden.

Read-Only

The Read-Only attribute makes a field read-only on the handheld, so that the user cannot enter any data in the field.

The uses for a Read-Only field are:

- ◆ For storing calculated results that you do not want the handheld user to directly modify. For example, if you are displaying the sum of three numbers in a field, you may not want the handheld user to be able to change the result except by changing one of the three numbers. The field storing the sum can be made Read-Only.
- ◆ For data filled in on the PC. If you are creating partially filled records on the PC, such as work orders, that the handheld user has to complete on the Palm device, you may want to protect the information that is generated on the PC.

To make a field Read-Only, check the Read-Only check box in the Advanced Field Properties window for the field.

Autodefault

The Autodefault feature sets the default value in a field to the value that was entered in that field in the previous record.

For example, imagine a field such as Employee on Duty, with options Mary and Bob. If Mary selects her name in that field, then every subsequent record that she enters will automatically have her name selected. If Bob then comes on duty and takes over using the handheld device, the first record that he creates will have Mary preselected. Bob then changes the response in the field to his name, and all subsequent records that he enters on his shift will have his name preselected.

Autodefault is a time-saving feature that is useful when a value does not change for a period of time. If you are counting inventory in different rooms, for instance, the room number will stay the same for tens or even hundreds of records, and you need to change the value in that field only when you move to another room.

To switch on the Autodefault feature, check the Autodefault check box in the Advanced Field Properties window. Note that:

- ◆ You can override the value entered in an Autodefault field at any time, by making another selection in that field. In the case of a Text, Numeric, or Currency field, select (highlight) the existing value and type a new one.
- ◆ Autodefault works with subforms, but only in a limited way. Autodefault records values from the previous record, but in a subform the previous record in the subform may be from a different parent record.
- ◆ Autodefault fields retain attributes such as whether they are read-only, required, or hidden. This is useful for scripts.

Display Key

The Display Key identifies the field that is used to display records when you are reviewing records on the handheld.

In Figure 6-2, the Display Key is a name field.

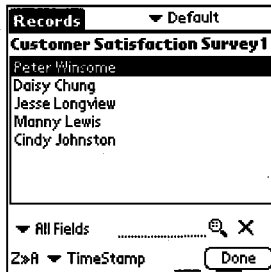


Figure 6-2: When you review existing records on the handheld, the Display Key field is shown for each record.

By default, the first field on a form is the Display Key.

- ◆ To select a different Display Key, check the Display Key check box in the Advanced Field Properties window.
- ◆ Only one field can be the Display Key. If you check the Display Key check box for more than one field, only the first field with this property selected will be the Display Key.

The Display Key field should be a field that is unique for each record, but this is not a requirement. A Text field or a Date field may be appropriate for use as a Display Key. A Pop-up List or Yes/No Checkbox may not be as appropriate as a Display Key, because several records can have the same value.

If the Display Key does not give you enough information to identify a record, you can display a second field in the list of records on the handheld, as shown in Figure 6-3. To display a second field, tap the arrow in the upper-right corner of the handheld screen and select a field.

Records	
▼ Date of Visit	
Customer Satisfaction Survey 1	
Peter Winsome	7/19/99
Daisy Chung	7/12/99
Jesse Longview	7/8/99
Manny Lewis	7/6/99
Cindy Johnston	7/5/99

▼ All Fields @ X

Z>A ▼ TimeStamp Done

Figure 6-3: Displaying a second field in addition to the Display Key field.

When using subforms, you can make the Display Key on the subform different from the Display Key on the parent form. For example, if the Display Key on the parent form is Customer Name, you may want to make the Display Key on the subform Date of Visit. That way, when you choose to review the subform records for a given customer, you will see a list of all visits.

Required

A Required field is one that the handheld user must enter data into before exiting the record.

Fields are made Required primarily to ensure that the handheld user does not accidentally omit entering a critical piece of information. To make a field Required, check the Required check box in the Advanced Field Properties window for the field.

When you exit a new record, the Forms program alerts the handheld user if Required fields have not been filled in. Figure 6-4 shows the message that appears on the handheld.

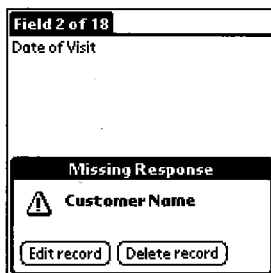


Figure 6-4: A message on the handheld alerts you that a Required field has not been completed.

The test of whether Required fields have been filled in occurs only when you exit the record. This means that if a Required field is in field 2, and 100 fields are on the form, you would receive the warning message only when you tried to exit after field 100.

An alternative to the handheld user's receiving a warning only when the user exits the record is to design the form to generate the warning message when the user tries to exit the Required field. That way, the handheld user will receive immediate feedback that the field should be filled, and can enter data in the field at once.



To set up this option, refer to Chapter 8, "Advanced Form Properties," under the topic "Field Level Validation."



A script can be used to make a Required field optional, or an optional field Required. Refer to Chapter 11, "Using Scripts."



Don't make a Hidden field Required, because the handheld user will not be able to access the field in order to enter data.

Primary Key

A Primary Key is a field or combination of fields that uniquely identifies a record. If more than one field is used to define a Primary Key, the combination of the primary key fields must be unique for each record.

In a form such as a patient list, an example of a single-field Primary Key might be a Social Security number. In a form such as a patient visit log, in which each patient may visit more than once, a multifield primary key can be used, such as a Social Security number plus the Date and Time of Visit.

When you exit a new record on the handheld, Pendragon Forms checks that the Primary Key field(s) are filled in and are unique, meaning that no two records on the Palm have the same primary key. If the primary key fields are not unique, you will be prompted to make them unique in order to save the record.

If no Primary Key is selected, Pendragon Forms creates a default Primary Key consisting of the UnitID, UserName, and TimeStamp fields that are automatically generated whenever a new record is created. The UserName is the Palm user name, and the TimeStamp is the creation date and time of the record. The UnitID defaults to a value of zero, and is present for backward compatibility with earlier versions of Pendragon Forms.

Selecting a Primary Key

In many cases, the default Primary Key is sufficient. However, there are some notable exceptions in which you may prefer to select your own primary key:

- ◆ If several handhelds need to share the same records. In Pendragon Forms, each time a record is modified, the handheld user name is assigned to the UserName field of the record. Because the default Primary Key uses the handheld UserName, there will be a duplicate record if a record is modified on a handheld with a different UserName. If multiple users need to share the same record, you can select a different Primary Key than the default. Refer to Chapter 16, "Working with the Palm VII," for more information on working in a multiuser environment.

- ◆ If you are linking to an external Microsoft Access or ODBC database. In order to link to an external database table, Pendragon Forms needs the primary key on your form to be the same as the primary key in your database table. Refer to Chapters 13 and 14 for information on linking to external databases.
- ◆ If there is a field that you want to be unique on the handheld, such as a bar code number in the case of an inventory application, or a Social Security number in the case of a patient tracking application.

To make one or more fields primary key fields, check the Primary Key check box in the Advanced Field Properties window for each field. You should make a Primary Key also a Required field, to ensure that the handheld user enters data in the field.



Once a form is frozen, the Primary Key should not be changed. This is because the underlying Primary Key in the database table is fixed when the form is frozen.

Working with Primary Keys on the Handheld

On the handheld, fields that are Primary Key fields appear with an icon of a key, as shown in Figure 6-5.

6374	
Customer Account#	6374
Customer Name	Corporate Boxes
Address	1554 E. Aaron St.
Region	↓South
Contact Name	Fred Martinsen
Contact Phone Nu	600-555-1010
Last Purchase Dat	8/15/99
Account Rep	Julie
<input type="button" value="End"/> <input type="button" value="←"/> <input type="button" value="→"/>	

Figure 6-5: Primary Key fields are identified with an icon of a key.

If you create a new record whose Primary Key matches that of an existing record on the handheld, when you attempt to exit the record you will see a message alerting you to the duplicate situation. Figure 6-6 shows the Primary Key message on the handheld.

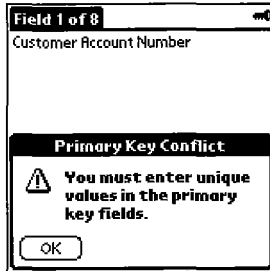


Figure 6-6: If you create a record with the same Primary Key as an existing record on the handheld, an error message will be displayed.

How Is a Primary Key Used during Synchronization?

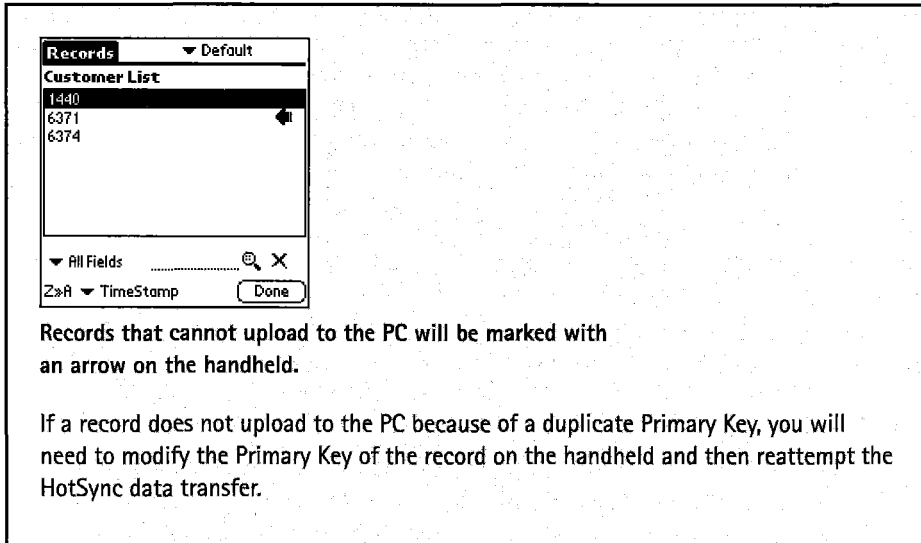
During a HotSync data transfer, Pendragon Forms checks the Primary Key of each record.

For existing records on the handheld, if a record has been modified, then the changes will update the record with the corresponding Primary Key in the database on the PC.

If a new record has been created on the handheld, the Primary Key should not match any other record already in the database. If the Primary Key is found to be unique, the new record is added to the database.

If a new record on the handheld has a Primary Key that matches an existing record in the database, the record will not be uploaded to the PC. This is to prevent a new record from accidentally overwriting an existing record – for example, in the case where the wrong Customer Account number is entered on the handheld, and the incorrect number matches an existing customer record. If you want new records on the handheld to update existing records on the PC, refer to Chapter 8, "Advanced Form Properties."

If a record cannot be uploaded to the PC during synchronization because of a duplicate Primary Key, an error message will be generated during the HotSync process to alert you. If you review the records for that form on the handheld, any records that could not be uploaded will be marked with an arrow, as shown in the accompanying figure.



Do Not Upload to PC

The Do Not Upload to PC check box enables you to select whether data in a field is sent to the PC during the HotSync process, or not. The default is to leave this check box blank, so that all fields upload to the PC.

Generally, you will want all fields to be uploaded to the PC in order to have a backup of your data on the PC. However, some special cases arise in which it is preferable not to upload data in a field to the PC.

- ◆ If you are linking to an external Access database that contains an AutoNumber field. An AutoNumber field in Microsoft Access is a field in which a new, unique number is created each time a new record is created. If you are creating records on the handheld, you may want to generate a temporary number on the Palm device but then leave the field empty when records are sent to the PC, so that the Access database can generate a number in the right sequence for that field. In this scenario, you would make the field not able to upload to the PC.
- ◆ If there are fields that you want to update only on the PC. The rule during synchronization is that an existing record that has been modified on the Palm device will update the corresponding record on the PC. In certain fields, if the PC always contains the most up-to-date information, then you should not allow the Palm device to overwrite the PC for these fields.

For example, imagine that you are creating work orders on the PC to send to the handheld, and you create a deadline date of Wednesday in a record. The handheld user performs a HotSync and receives the record. The deadline date is then changed to Friday on the PC. In the meantime, the handheld user modifies the record to indicate that work has begun. When the handheld synchronizes, the modified record overwrites the record on the PC, which means that the deadline date is reset to Wednesday. To prevent the handheld from overwriting the deadline date with older information, you can set this field to not upload to the PC.

To make a field not upload to the PC during synchronization, check the Do Not Upload to PC check box in the Advanced Field Properties window for the field.



The Do Not Upload to PC property is designed to protect AutoNumber fields on the PC and to protect fields that already contain data on the PC. In the case where a record is new on the handheld and does not match any records on the PC, all fields (except AutoNumber fields) will upload to the PC even with this property switched on.

Non-Printing

When you are printing directly from the handheld, the Non-Printing option enables you to select whether a field will be printed or not.

If you are printing records to create a hard copy backup, you should print all fields. However, if you are printing records for customers to have a copy, there may be fields that you do not need to print. For example, on a work order, the name of the dispatcher who received the phone order may be useful for the handheld user but of no relevance to the customer who receives a copy of the work performed.

In the Advanced Field Properties window, check the Non-Printing check box to mark a field as not to be printed. Leave this check box blank for all fields that are to be printed.

Max and Min

The Max and Min options work with Numeric fields only and enable you to specify a numeric range for data entered in a specific field on the handheld. The Max value is the upper limit of the range, and the Min value is the lower limit of the range.

The largest number that you can specify for Max or Min is 15 digits. For a negative number, include a minus sign (in Graffiti, a hyphen) before the number.

You can also choose to enter Max and Min values as exponents.

- ◆ If you enter a value for Max, you should also enter a value for Min; otherwise, the program defaults to a minimum of zero.
- ◆ The actual values of Max and Min are included in the range. For example, if a numeric range is specified from 0 to 100, the handheld user can enter zero as a valid number in the range or enter 100 as a valid number in the range.

On the handheld, you may want to state in the field name what is the expected numeric range of the user's response. If you enter a number outside the range, a message, as shown in Figure 6-7, will be displayed when the handheld user tries to leave the field.

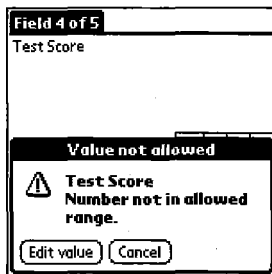


Figure 6-7: A message alerts the handheld user if a value outside of the numeric range specified by the Max and Min Advanced Field Properties is entered.



In addition to setting a numeric range with Max and Min, you can also check the Integer property, so that only whole numbers within the range can be entered.

Max Length

The Max Length property applies only to Text fields; it specifies the maximum number of characters that the handheld user can enter in a field.

If the Max Length field is left blank, the default maximum of a Text field is 255 characters. This is the maximum length that a Text field can be in Microsoft Access and still be capable of being sorted.

You can set the Max Length to a number from 1 to 2000. Microsoft Access is very efficient at storing data, so unless you need responses longer than 255 characters, leave the default Max Length.

If you make the Max Length more than 255 characters, then internally the field will be stored as an Access Memo field, not a Text field. Memo fields cannot be sorted in Access.



When you freeze a form, the Max Length property is used to set the maximum number of characters allowed in the corresponding Text field in the database table. After the form is frozen, you cannot change the Max Length.

The maximum size of a record on the handheld is 64KB. This means that if you want to capture 2,000 characters per Text field, the entire form cannot exceed approximately 30 Text fields. Realistically, it is unlikely that the handheld user will need to enter 2,000 characters in every Text field.

Lookup List

The Lookup List option enables you to select a Lookup List to be used with a Text field.

With a regular Lookup List field, you can select only one item from the list. With a Lookup List attached to a Text field, however, the handheld user can repeatedly select items from the list to copy into the Text field. The user can also add any text to the Text field.

Uses of a Lookup List with a Text field include:

- ◆ Keeping a list of common phrases, to speed up data entry on the handheld. For example, physicians who use standard phrases to describe a patient's condition may choose to create a Lookup List containing the phrases. A selection from the list is equivalent to writing an entire sentence.
- ◆ A list of commonly used bar codes, useful for some bar code scanning applications. If there is a problem with the scanning mechanism, you can select an item from the list instead.

To select a Lookup List to be used with a Text field, display the Advanced Field Properties window and select a Lookup List in the Lookup List field. You can change which Lookup List is used with the Text field even after the form is frozen.

On the handheld, when you display the Text field in Field View, as shown in Figure 6-8, a Lookup List icon is displayed on screen. Tap this icon to display the

Lookup List. After selecting an item from the list, you can tap the Lookup icon again if you need to select more items to copy into the Text field.

Lookup List icon

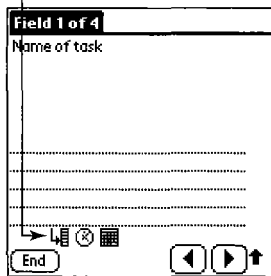


Figure 6-8: A Lookup List icon is displayed when you associate a Lookup List with a Text field.

Pattern

The Pattern option applies only to Text fields. By selecting this option, you can specify the type of characters that the handheld user can enter in the field.

The Pattern options are:

- ◆ **Alpha:** alphabetic characters (A to Z) only
- ◆ **Alphanumeric:** alphabetic characters (A to Z) or numbers (0 to 9)
- ◆ **Digits:** numbers (0 to 9) only
- ◆ **Printable characters:** no tabs or carriage returns
- ◆ **Uppercase:** such that when the handheld user exits the field, letters are converted to all uppercase characters
- ◆ **Lowercase:** such that when the handheld user exits the field, letters are converted to all lowercase letters

To set the Pattern option, display the Advanced Field Properties window for a Text field, and then select an option in the Pattern field.

If the handheld user enters a character that is not in the pattern, when he or she attempts to leave the field, a message will be displayed, as shown in Figure 6-9. You may want to use the field name to give the handheld user instructions about the expected pattern.

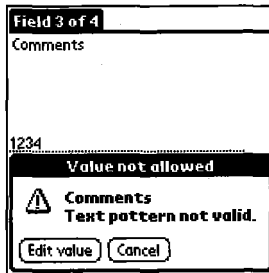


Figure 6-9: An error message is displayed if the handheld user does not enter characters that conform to the selected Pattern property.



If you specify a character pattern, no characters outside that pattern are allowed. For example, if you select Digits, you can enter 1234567, but you cannot add a hyphen, such as 123-4567, or put a space between digits, such as 123 4567.

Summary

In this chapter you saw how to set Advanced Field Properties on a field-by-field basis. Advanced Field Properties can be used to make a field required or hidden, and to specify a range in a Numeric field, or the maximum length of a Text field. Selecting the Display Key field or selecting a Primary Key is also performed in the Advanced Field Properties window.

Chapter 7

Synchronization Rules

IN THIS CHAPTER

- ◆ Understanding how the HotSync Manager Works
- ◆ Using the Pendragon Forms Conduit
- ◆ Understanding the Pendragon Forms Synchronization Process
- ◆ Setting up Users and User Groups
- ◆ Deleting Forms
- ◆ Switching Off Synchronizing Forms

PENDRAGON FORMS SUPPORTS bi-directional synchronization, meaning that during synchronization, new records on the handheld are sent to the PC, and new records on the PC are sent to the handheld.

The program is designed for central management on the PC, and so the selection of which forms and data go to which handheld is controlled on the PC.

The Pendragon Forms Conduit

A conduit is a plug-in software module that acts as a communications bridge between the PC and a Palm Computing device. Each conduit extends the functionality of the HotSync Manager software to synchronize a specific application.

The Pendragon Forms conduit runs during the HotSync process to synchronize form designs and records between the PC and the handheld.

The Pendragon Forms conduit file is Forms3.dll (or Forms3a.dll if you are using Microsoft Access 2000) and is stored in the Windows system folder on your PC. If you have Windows 9x, the system folder is C:\Windows\System. If you are using Windows NT, the system folder is C:\WINNT\System32.

How Does the HotSync Manager Work?

The HotSync Manager software that ships with your Palm organizer manages the communication between your handheld device and your PC. This communication can be a direct serial port connection, a modem connection, or a network connection.

During installation of the Palm Desktop software, a shortcut to the HotSync Manager is placed in your Startup group so that it will run every time you start your computer. (To change this, simply delete the shortcut from C:\Windows\Start Menu\Programs\Startup.) The HotSync Manager listens for requests for communications from the handheld devices.

When you click the HotSync button on your cradle, a request is sent to the HotSync Manager to begin a HotSync data transfer. The HotSync Manager first determines the user name assigned to the handheld and then downloads a list of the applications that are installed. For each application installed on the handheld, the HotSync Manager searches for an installed conduit that is capable of synchronizing the application.

The built-in applications, such as the Date Book, Address Book, Memo Pad, and To Do List, each have conduits installed with the HotSync Manager software. The Pendragon Forms conduit is installed when you first install Pendragon Forms. .

After the HotSync Manager has determined which applications have conduits, and which applications will synchronize, it passes control of the synchronization process to each conduit in turn. Each conduit synchronizes data associated with a particular application. As a conduit takes control of the synchronization process, you will see the HotSync Progress dialog update to show you which application is synchronizing.

Internally, each conduit is managing the flow of data between the PC and the handheld, fetching and storing records as appropriate.

When Does a Conduit Synchronize?

Three basic conditions need to be met before a given conduit will synchronize with a handheld. First, the relevant Palm OS application must already be installed on the handheld. This means that you must first perform a HotSync data transfer to install the Pendragon Forms Palm OS application (the Forms3.prc file) before you will see any form designs or data appear on the handheld.

Second, the conduit file must be present on the PC. The Pendragon Forms conduit file is Forms3.dll (or Forms3a.dll if you are using Microsoft Access 2000), and it is normally installed into your Windows system directory (C:\Windows\System in Windows 95 and Windows 98, and C:\Winnt\System32 in Windows NT).

Finally, the appropriate entries must be made in the Windows Registry. These settings are used to tell the HotSync Manager software where the conduit file can be found, the name of the application, and which applications the conduit will synchronize.

The Pendragon Forms setup program will automatically set up the Registry entries and install the conduit file. You can install the Forms3.prc file by clicking Start → Programs → Pendragon Forms 3.0 → Install Forms 3.0 on Handheld

The Synchronization Process

When the HotSync Manager gives control of the synchronization process to the Pendragon Forms conduit, the conduit executes the following sequence:

1. All form designs, records, and Lookup Lists are uploaded from the handheld into the desktop PC's memory.
2. The Forms3.mdb database file (or Forms32k.mdb if you have Microsoft Access 2000) is opened and the handheld user is verified as an active user, authorized to synchronize Pendragon Forms.
3. For each form that should be synchronized with the handheld:
 - The database on the desktop is opened. (This may be an external database.)
 - Changes are uploaded to the database. New records from the handheld are inserted into the database, and changed records from the handheld update (overwrite) existing database records. If a changed record on the handheld cannot find an existing record on the desktop, an insert into the database is performed.
 - All records on the handheld are marked for deletion, except records that were not able to be placed into the database.
 - If the form design is new or has been modified and redistributed, it is marked for installation to the handheld.
 - A query is run to determine which records should be sent to the handheld. Records are selected based on form properties and download criteria.
 - Forms that are to be deleted, along with their records, are marked for deletion on the handheld.

4. The Last HotSync date is updated in the User List on the desktop.
5. Deleted form designs and records are removed from the handheld. Form designs, records, and Lookup Lists are sent from the desktop to the handheld.

The User List

The Pendragon Forms Manager maintains a User List of active handheld units that can synchronize with the database during the HotSync process.

In a single-user scenario, only one handheld can synchronize with the Pendragon Forms database.



When you first install Pendragon Forms, you are prompted to enter your handheld user name. This user name is added to the User List as the single user.

If a multiuser license is activated, you can add additional handheld user names by clicking the Users button in the Pendragon Forms Manager. The User List is displayed (see Figure 7-1).

Adding Handheld Users

To add a user to the User List, type the handheld user name in a blank row of the User List. Check the Active box to make the user active. To save the record, click the pencil icon (not shown in the figure) to the left of the row, until it changes to a solid black triangle (shown in figure). Alternatively, you can press Tab until the cursor moves to the next row.

The user name that you type in the User List must exactly match the handheld user name. If you do not know the handheld user name for a particular device, tap the Applications button and then tap the HotSync icon on the handheld. The user name will be displayed in the upper-right corner of the screen, as shown in Figure 7-2.

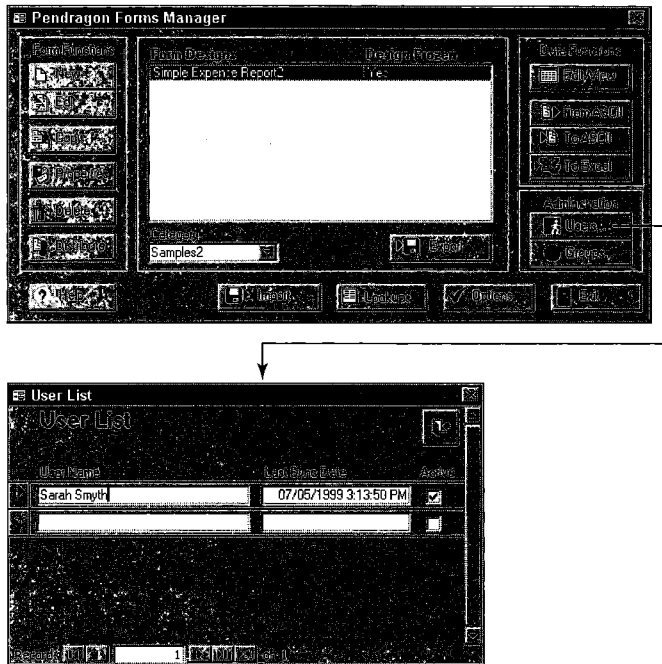


Figure 7-1: Clicking the Users button in the Pendragon Forms Manager window displays the User List.

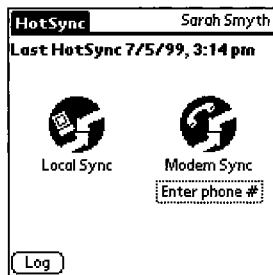


Figure 7-2: Tap the HotSync icon on the handheld to display the handheld user name.

How Does the Handheld Device Get a User Name?

The very first time that you perform a HotSync data transfer with a new Palm organizer, the HotSync Manager prompts you to enter a user name for the device. Pendragon Forms uses this handheld user name to determine whether the handheld can synchronize with the Forms3.mdb database (or Forms32k.mdb if you have Microsoft Access 2000).

If you need to change the handheld user name of a Palm organizer — for example, if you assigned one user name to the handheld for test purposes, and you now need to assign the handheld to a person — open the Palm Desktop software by clicking Start → Programs → Palm Desktop → Palm Desktop. In the Palm Desktop application, click Tools → Users. In the list of users, click a name and click the Rename button. Type a new user name for the Palm device, and then perform a HotSync data transfer.

If you change the user name of a Palm device, update the user name in the Pendragon Forms Manager User List also.

Deleting Users

If a handheld unit is no longer being used to synchronize Pendragon Forms, you can uncheck the Active box in the User List to make the user inactive. If an inactive user performs a HotSync data transfer, the Pendragon Forms application will not synchronize for that user.

Alternatively, you can delete the user from the User List. As shown in Figure 7-3, click the gray cell to the left of the row containing the user name, to select the row. Then press the Delete key on the keyboard.

Selected row

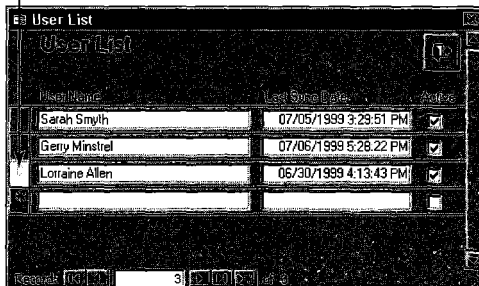


Figure 7-3: To delete a user from the User List, select a row and then press the Delete key.

User Groups

User Groups are the mechanism by which Pendragon Forms determines which form designs and data are to be sent to which handheld device.

During the HotSync process, a handheld automatically receives any forms that are assigned to the User Group to which the handheld belongs. If a form is removed from a User Group, the form will be removed from the handheld during the next synchronization.

In the Pendragon Forms Manager window, click the Groups button. The User Groups window is displayed (see Figure 7-4).

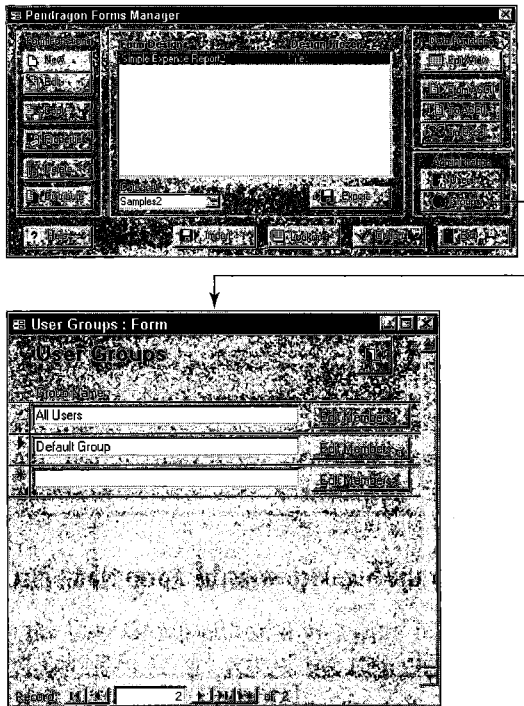


Figure 7-4: Clicking the Groups button in the Pendragon Forms Manager displays the User Groups window.

Two special User Groups are predefined:

- ◆ **Default Group.** When you click the Distribute button in the Forms Manager to send a form to the handheld, the form is added to the Default Group. Handheld users in the Default Group receive the form during their next HotSync data transfer. In a single-user scenario, the handheld user

name that you enter when Pendragon Forms is installed is added as the user in the Default Group. This means that when you distribute a form, the form is queued to be sent to your handheld.

In a multiuser scenario, it is a good idea to use the Default Group for yourself as the developer, but then create separate User Groups when you are ready to distribute a form to other handhelds.

- ◆ **All Users Group.** If a form is added to the All Users Group, then all active users in the User List receive the form during their next HotSync data transfer. The All Users Group is a quick way to distribute a form to all active users without having to create additional User Groups.

Creating a User Group

You may want to create your own custom User Group to allocate specific form designs to a group of handheld devices. For example, you can create different groups for people in different departments.

In the User Groups window, type a name for the User Group in a blank row. Click the Edit Members button to display the User Group Editor window (see Figure 7-5).

To select which handheld users belong to the User Group, click the arrow in the blank row of the Username list and select a handheld user name.



Only users from the User List can be selected for membership to a group.

To assign a form to a User Group, click in the blank row in the Form Name list, and select a form.



Only forms that are frozen and have been distributed appear in the list. To distribute a form, select the form in the Forms Manager and click the Distribute button.



You can assign a form design to more than one User Group.

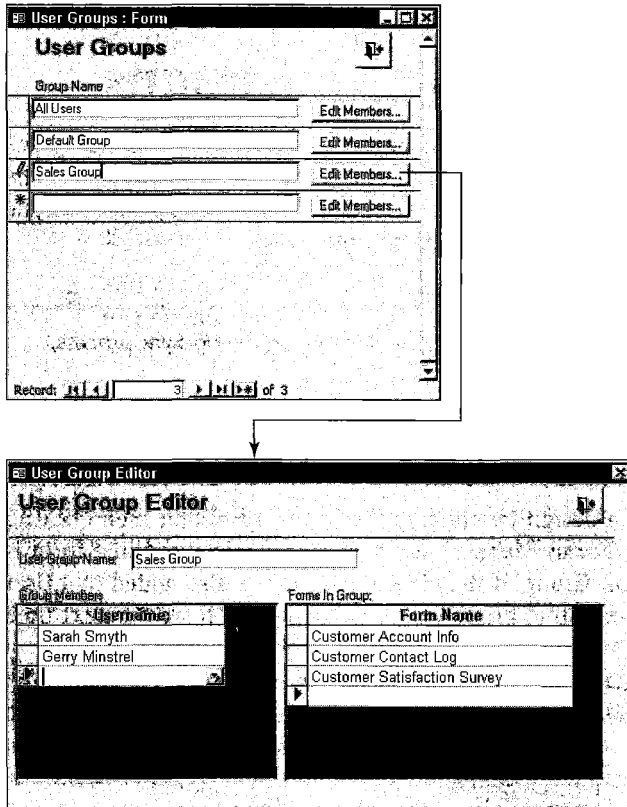


Figure 7-5: To create a new User Group, type a name in a blank row and then click the Edit Members button to display the User Group Editor window.

Deleting a User or a Form from a User Group

In the User Group Editor, if you want to remove a handheld user from a group, click the gray cell to the left of the row containing the handheld user name. Press the Delete key on the keyboard.

Similarly, to remove a form design from a group, click the gray cell to the left of the row containing the form name and then press the Delete key on the keyboard. Deleting a form from a group means that when each of the members in the group performs a HotSync data transfer, the form design and all its records will be removed from the handheld.

User Groups and the Synchronization of Form Designs

During the synchronization process, the Pendragon Forms conduit checks to see whether the handheld user name is listed as an Active user in the User List. If the handheld user is not active, an error message will be generated during the HotSync data transfer, and Pendragon Forms will not synchronize.



For information on error messages generated during the HotSync process, see Appendix A, "Troubleshooting Tips."

You distribute a new form design by clicking the frozen form and then clicking the Distribute button in the Forms Manager window. Once distributed, the form is sent to all users in the Default Group. If the distributed form is also added to a User Group, all the handheld users in that group will receive the new form the next time they synchronize.

If a frozen form is modified, you can redistribute it by clicking the name of the form, and then clicking the Distribute button in the Forms Manager window. When handheld users synchronize, the conduit compares their last HotSync date with the date of redistribution of the form. If the last HotSync date is before the redistribution date, the updated form design will be sent to the handheld.

During synchronization, handheld devices with forms that have been deleted from a User Group will first have their data uploaded to the PC, and then the form design and data will be removed from the handheld. Deleting forms requires some care, as explained in the following section.

Deleting Form Designs and Data

Pendragon Forms is designed for central control from the desktop PC. This means that the deletion of forms and records from the handheld is controlled via the desktop.

Some care must be taken when deleting forms, as the following scenarios illustrate:

- ◆ If you delete a form from the PC without removing the form from the User Groups, the form will remain on the handheld.

- ◆ If you delete a form from the handheld without removing the form from the User Groups, the form will return to the handheld during the next synchronization.

The proper sequence for deleting forms is as follows:

1. Delete the form from any User Groups to which the form belongs. In a single-user installation, the Default Group contains the form design.
2. Perform a HotSync data transfer. This will remove the form design and the records for that form from the handheld. In a multiuser scenario, you will need to wait until all users have had an opportunity to synchronize.

After the form has been removed from all the handheld devices, you can choose to keep the form and its data intact in the Pendragon Forms Manager. Alternatively, if you are sure that you do not need the form or its records, you can delete the entire database table by clicking the name of the form in the Forms Manager, and then clicking the Delete button in the Forms Manager window.



Each form has a unique Form ID number that is used to synchronize the form with the handheld. If you accidentally delete a form from the PC but you still have the form on your handheld and still need to synchronize the form, you will need to attempt to recover the form design in order to synchronize. Refer to Chapter 9, "Managing Data on the PC," for information on your recovery options.

Deleting the Pendragon Forms Application from the Handheld

If you no longer need to have Pendragon Forms installed on your handheld, you can delete the Pendragon Forms application (Forms3.prc) from the Palm organizer to free up memory on the handheld. Deleting the Forms3.prc file will also remove all of your form designs and data in those forms from the handheld.

Before you delete the Forms application from the handheld, perform a HotSync data transfer to back up any records that are on the handheld. Tap the Applications button on the handheld, and then tap the Menu button. On the App menu, tap the Delete option (or use the Graffiti shortcut /D). A list of applications that can be deleted will be displayed. Tap the Forms 3.0 application, and then tap the Delete button.

Limiting the Number of Records on the Handheld

Because the database on the PC can store many more records than the handheld, one of the key functions of the Pendragon Forms conduit during synchronization is to determine which records should be removed from the handheld.

In the Pendragon Forms Manager, the Form Properties window enables you to specify the Data Persistence options that determine how long a record will remain on the handheld. Select a form and then click the Properties button to view the Form Properties window, shown in Figure 7-6.

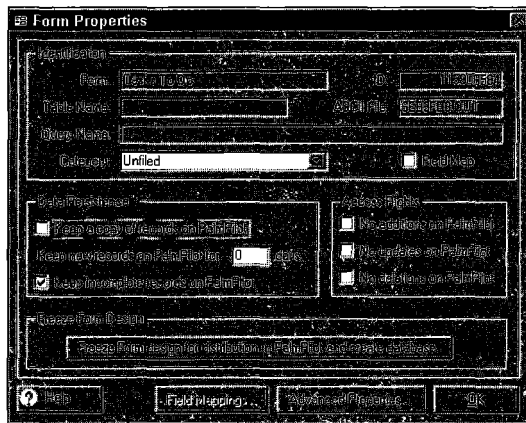


Figure 7-6: The Form Properties window

The Data Persistence options are as follows:

- ◆ To remove all records after a HotSync data transfer, leave all Data Persistence options blank.
- ◆ To keep a copy of the records on the handheld, check the box Keep a Copy of Records on PalmPilot.
- ◆ To keep records on the handheld for a specified number of days, enter a number from 0 to 999.
- ◆ To remove records when a Completion Checkbox field on the form is checked, select the option Keep Incomplete Records on PalmPilot.

Only one of the Data Persistence options can be selected at a time. You can change a Data Persistence option after a form has been frozen, and then redistribute

your form and perform a HotSync data transfer for the changes to take effect on the handheld..

Several form properties override the Data Persistence options. In the Form Properties window, click the Advanced Properties button to access the Advanced Form Properties window, as shown in Figure 7-7.

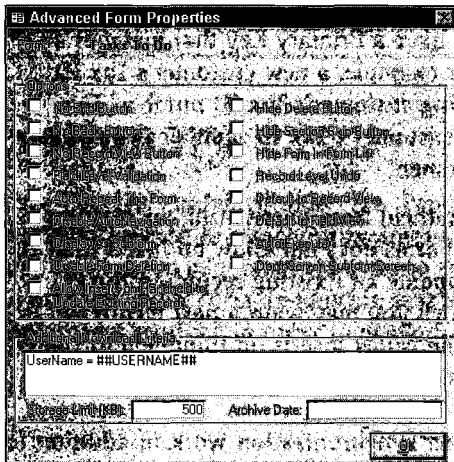


Figure 7-7: The Advanced Form Properties window

Advanced Form Properties are covered in detail in Chapter 8. However, it worth noting here the options that affect synchronization:

- ◆ **Storage limit.** To protect the handheld from receiving too many records, regardless of the Data Persistence option, a default storage limit of 500K is set. If you have a large number of records to keep on the handheld, and you know that the handheld has sufficient free memory, you can increase the Storage Limit.

If you have more records on the PC than can fit on the handheld, records will be placed on the handheld up to the Storage Limit. Records will be placed on the handheld in ascending order of the Primary Key. The default Primary Key uses a creation TimeStamp, which means that the first 500KB of records, starting from the earliest record, will be placed on the handheld.

- ◆ **Archive Date.** If you do not want to receive the earliest records on the handheld, you can periodically set an Archive Date. Records before the Archive Date will not be placed on the handheld. For example, an Archive date of 09/30/99 will remove all records from before September 30, 1999.

- ◆ **Additional Download Criteria.** Additional Download Criteria enables you to fine-tune which records are sent to the handheld. The default value, `UserName = ##USERNAME##`, means that only records in which the `UserName` field matches the handheld user name are sent to the handheld. You can choose your own download criteria, as described in Chapter 8, “Advanced Form Properties.”
- ◆ **Allow Inserts on Handheld to Update Existing Records.** Generally, a new record on the handheld should not have a Primary Key that matches an existing record in the database. For example, a new customer’s Tax ID number should not match an existing customer’s Tax ID number. This implies a conflict – either the new customer’s Tax ID number is incorrect, or the individual is not a new customer.

To prevent a new record on the handheld from overwriting an existing record in the database, Pendragon Forms does not allow an Insert (that is, a new record) on the handheld to update an Existing record on the PC. There are rare circumstances in which you may want a new record to overwrite an existing record, and the Allow Inserts on Handheld to Update Existing Records option allows this.

The Advanced Form Properties that affect synchronization work in addition to the Data Persistence options that you select. For instance, if you choose the Data Persistence option to Keep Incomplete Records on PalmPilot, and you also choose an Archive Date of today, only the incomplete records starting from today will be sent to your handheld.

Switching Off Synchronizing Forms

You can use the HotSync Manager to switch off synchronizing Pendragon Forms. You may want to do this if, for example, you need to quickly synchronize one or two applications, and Pendragon Forms is not needed.

To switch on or off synchronizing Pendragon Forms, right-click the HotSync Manager icon (the icon with red and blue arrows) in the system tray of your Windows Taskbar. Choose Custom from the menu.

In the Custom window, select your handheld user name. In the list of applications on your handheld, click Pendragon Forms to select it and then click the Change button.

- ◆ Select the Do Nothing option to turn off synchronizing Pendragon Forms on the next synchronization only. Check the Set as Default box if you want to switch off synchronizing Pendragon Forms indefinitely.

- ◆ Select the Synchronize option and check the Set as Default box to return to synchronizing Pendragon Forms during every HotSync data transfer.



If you switch off synchronizing Pendragon Forms, you will not have a backup of any new records that you create on the handheld.

Summary

This chapter explained how the Pendragon Forms conduit works during the HotSync process to synchronize forms and data between handheld and PC. You also saw how to add handheld users to the User List, and how to create User Groups to manage which form designs are sent to which handheld. In addition, you learned the process required in order to delete form designs and data from the handheld.

Chapter 8

Advanced Form Properties

IN THIS CHAPTER

- ◆ Controlling navigation through a form
- ◆ Controlling data entry
- ◆ Automating the creation of new records
- ◆ Controlling the deletion of forms designs
- ◆ Understanding advanced synchronization rules

ADVANCED FORM PROPERTIES are features that affect an entire form. You can set Advanced Form Properties in order to gain more control over the way that a form is used on the handheld.

Advanced Form Properties are grouped by function in Table 8-1.

TABLE 8-1 ADVANCED FORM PROPERTIES

Category	Advanced Form Properties
Navigation through a form	No End Button – hides the End button in Field View No Back Button – prevents the user from going back to a previously answered question in Field View No Record View Button – prevents the user from accessing the form in Record View Hide Section Skip Button – prevents the user from skipping over a section of a form Disable AutoNavigation – switches off the AutoNavigate feature Default to Field View – causes new records to be displayed in Field View

Continued

TABLE 8-1 ADVANCED FORM PROPERTIES (Continued)

Category	Advanced Form Properties
Navigation through a form (continued)	Default to Record View – causes new records to be displayed in Record View
Data Entry	Field Level Validation – checks that a required field has been filled in as user moves to the next field on the form Record Level Undo – gives the user the option to cancel modifications to an existing record
Subforms	Display as Subform – displays subforms on the handheld with a different icon than regular forms Hide Form in Form List – causes a form design not to be displayed in the list of forms on the handheld Don't Sort on Subform Screen – causes subform records not to be sorted
Automating the Creation of a New Record	Auto-Repeat This Form – causes a new record to be created as soon as the user ends one record Auto Execute – creates a new record for a form as soon as the Forms icon is tapped
Deletion	Disable Form Deletion – prevents a form design from being deleted from the handheld Hide Delete Button – causes the Delete button in the Forms List to be hidden when a given form is selected, to minimize the risk of accidental deletion of the form design
Advanced Synchronization Rules	Allow Inserts on Handheld to Update Existing Records – allows a new record on the handheld to overwrite an existing record on the PC Storage Limit – sets a size limit in kilobytes for the number of records that can be maintained on the handheld for a given form design Archive Date – causes all records that were created before the archive date to be removed from the handheld Additional Download Criteria – sets criteria for selecting which records are sent to the handheld

In the Pendragon Forms Manager, select a form design and then click the Properties button to access the Form Properties window. In the Form Properties window, click the Advanced Properties button to display the Advanced Form Properties window (see Figure 8-1).

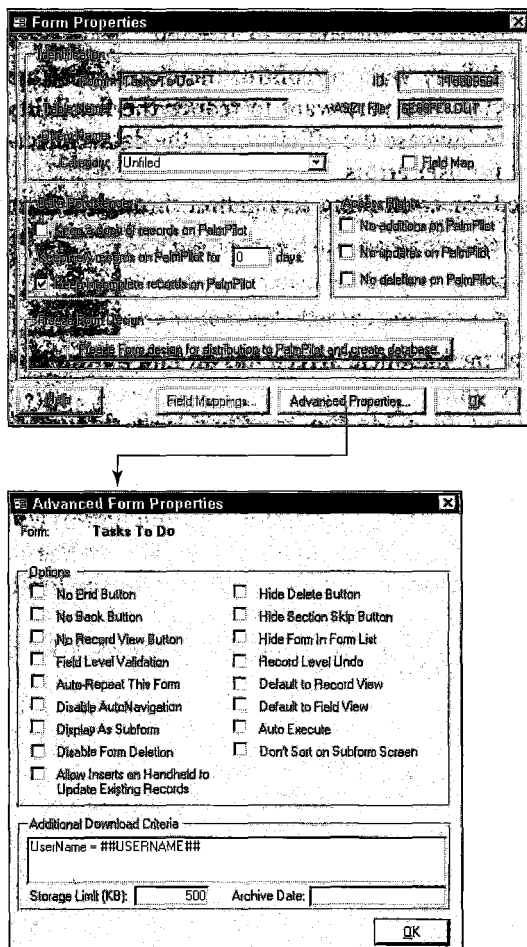


Figure 8-1: Clicking the Advanced Properties button in the Form Properties window displays the Advanced Form Properties window.



You can change Advanced Form Properties after a form has been frozen. After making your changes, redistribute the form and perform a HotSync data transfer for your changes to take effect on the handheld.

Controlling Navigation through a Form

Pendragon Forms was designed to give you a lot of flexibility to move forward and backward through the fields on a form, to switch from Field View to Record View at any time, and to end a record at any time.

In certain applications, however, you may want to control the experience that the handheld user has on the Palm device. You may want to reduce the flexibility of navigating through a form in order to increase the efficiency of the data-collecting process, or to reduce the potential for user errors.

The Advanced Form Properties used for controlling navigation through a form are described here.

No End Button

In Pendragon Forms, the default is that the handheld screen has an End button that enables the user to exit a record at any time.

In some applications, you may want to guide the handheld user through a form and not allow him or her to exit the form before going through every field on the form.

If you check the Advanced Form Property of No End Button, the End button will not be displayed on the handheld screen in Field View. The handheld user has to use the Next button (right arrow button) to step through every field on the form. Tapping the Next button in the last field on the form exits the record.

Figure 8-2 shows a form with the End button, and Figure 8-3 shows a form without the End button.

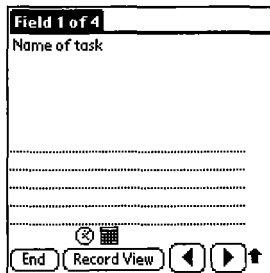


Figure 8-2: By default, every form has an End button.

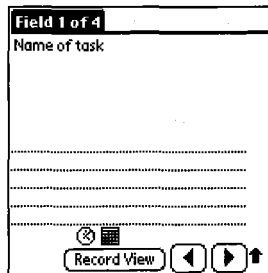


Figure 8-3: A form with the Advanced Form Property of No End Button

Removing the End button does not ensure that the handheld user fills in every field; it just makes the user step through every field. If you want to ensure that a field is actually filled in, set the Advanced Field Property of Required.



Hiding the End button affects only Field View, not Record View. For this form property to be effective, you should also set the Advanced Form Property of No Record View Button, so that the handheld user is confined to using Field View.



If the handheld user performs a HotSync before completing a record, the incomplete record will be sent to the PC. In this scenario, hiding the End button does not offer any protection from an incomplete record.

No Back Button

The Back button or left arrow button that is visible in Field View enables the handheld user to move backward to previous fields on a form. This means that the user can enter data in a field, and then before ending a form, he or she can go back and change an entry in an earlier field.

In some form designs, you may want the handheld user to enter data in a field and be able to go only forward through the form, not back to change the response.

If you check the No Back Button option in the Advanced Form Properties window, the handheld user will be able to move only forward through the form in Field View.

Figure 8-4 shows a form with the Back button, and Figure 8-5 shows a form with the Back button hidden.

Figure 8-4: A form with the Back button

Figure 8-5: A form with the Back button hidden

If you do not want the handheld user to go back to previous fields, you should also prevent the user from accessing Record View, by setting the Advanced Form Property of No Record View Button.



TIP Hiding the Back button and the Record View button is effective only in preventing the handheld user from changing responses in a new record. It is still possible to review an existing record and make changes there. If you do not want the handheld user to be able to make changes by reviewing a record, set the access right of No Updates on PalmPilot, in the Form Properties window.

No Record View Button

When you are in Field View, viewing one field on the handheld screen at a time, it is possible to switch to Record View by tapping the Record View button. Record View displays eleven fields on screen at a time, and from Record View it is possible to jump to any field on the form.

If you want to control the order in which the handheld user views fields on the form, you may want to hide the Record View button.

Hiding the Record View button is also recommended if you are using branching scripts, because branching scripts work only in Field View.

To hide the Record View button, check the No Record View button in the Advanced Form Properties window.

Figure 8-6 shows a form with the Record View button, and Figure 8-7 shows a form without the Record View button.

Figure 8-6: A form with the Record View button

Figure 8-7: A form without the Record View button

Hide Section Skip Button

If you are using Section fields on a form, note that in Field View the Section field has a Skip Section button that enables you to skip over a section and go to the start of the next section. This is useful if sections of the form are optional.

If sections of a form are not optional, you may prefer to hide the Skip Section button, so that the handheld user has to at least step through the fields in a section of the form.

To hide the skip button, check the option Hide Section Skip Button in the Advanced Form Properties window.

Figure 8-8 shows a Section field with the normal Skip Section button, and Figure 8-9 shows a Section field without the button.

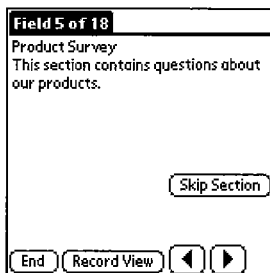


Figure 8-8: A normal Section field

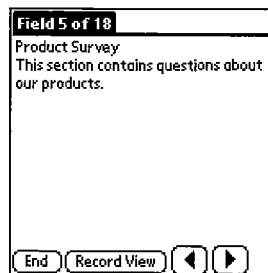


Figure 8-9: A Section field with the Skip Section button hidden



A Section field without a Skip button can make a good splash screen or title for forms that start in Field View

Disable AutoNavigation

The AutoNavigate feature is generally set on the handheld. In the Forms List screen that displays the forms on the handheld, an Auto check box switches AutoNavigate on or off. The AutoNavigate feature advances you to the next field on a form as soon as a selection is made in a selection field or date field.

AutoNavigate also automatically displays Popup Lists, Lookup Lists, and Date fields, obscuring the field name from the user.

In some form designs, you may need the handheld user to read the field name in order to obtain instructions on the data that should be entered in the field. In these

circumstances, it is better for AutoNavigate to be switched off, so that the handheld user can read the question and then pop up the list or calendar to make a selection.

The Advanced Form Property of Disable AutoNavigation ensures that whatever the AutoNavigate setting is on the handheld, when you select to create a record for this form, AutoNavigate will be switched off.

Check the Disable AutoNavigation check box in the Advanced Form Properties window to switch off AutoNavigate for the selected form.

Default to Field View and Default to Record View

On the handheld, the user can choose whether new records are created in Field View or in Record View. From the Forms List screen, if the user taps the handheld Menu button and selects the Options menu (or uses the Graffiti shortcut /V), the user can choose Toggle Default View to change the default view from Field View to Record View and vice versa.

- ◆ Check the Default to Field View check box in the Advanced Form Properties window if you want to ensure that a new record for the selected form is created in Field View.
- ◆ Check the Default to Record View check box in the Advanced Form Properties window if you want to ensure that a new record is created in Record View.

Default to Field View and Default to Record View are mutually exclusive—you cannot choose both options for the same form.

Controlling Data Entry

There are many ways to control data entry on the handheld. Chapter 6, “Advanced Field Properties,” covers the primary ways to exert control on a field-by-field basis.

Two Advanced Form Properties assist you with respect to the control of data entry.

Field Level Validation

If you need to ensure that the handheld user fills in a particular field, you can set the Advanced Field Property of Required. When the handheld user exits a record, an error message will be generated if some required fields are blank, and the user will have to fill in the Required fields before exiting the record.

Figure 8-10 shows the error message that is generated on the handheld when a Required field is missing.

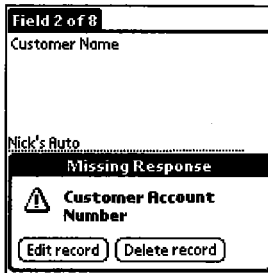


Figure 8-10: If a Required field is left blank, you will receive an error message on the handheld.

The check that a field is Required is performed only when the user attempts to exit a record. If the user is entering data in Field View, the program returns the user to the blank Required field. However, in Record View the program does not return the user to the Required field. If the Required field is Field 10, and the last field on the form is Field 90, it is cumbersome to navigate back to the Required Field in order to fill in a response.

The Advanced Field Property of Field Level Validation gives the handheld user feedback on whether a response is missing in a Required field as soon as the user leaves the field. Because the user receives the feedback on the missing response immediately, it is easier to correct the error than to wait until the user is exiting the record.

To switch on Field Level Validation, check the Field Level Validation check box in the Advanced Form Properties window.



If you are using numeric ranges in a Numeric field, you do not need to use Field Level Validation. Pendragon Forms automatically checks whether the number is in the valid range when the user leaves the Numeric field.

Record Level Undo

Record Level Undo gives the handheld user the opportunity to change his mind on whether to keep changes to an existing record.

In Pendragon Forms, when the handheld user is reviewing an existing record, he can make modifications to the record. These changes are saved by default when the user exits the record.

In some form designs, it is useful to allow the handheld user to decide whether changes should be saved or not. For example, if a user is completing required fields on forms that are partially filled on the PC, you may want to give the user the option to revert to the state a record was in before modifications were made.

In the Advanced Form Properties window, check the Record Level Undo check box to switch on the option of saving or discarding changes. If the user reviews an existing record, makes changes, and then taps the End button to exit the record, a dialog box as shown in Figure 8-11 will be displayed.

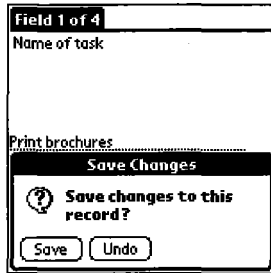


Figure 8-11: If Record Level Undo is switched on, you can choose to keep or discard changes to an existing record.

If the user taps Save, the changes to the record will be saved. If the user taps Undo, the changes will be discarded, and the record will revert to the way it was before the changes were made.

Controlling the Appearance of Subforms

Advanced Form Properties can be used to affect how or if a subform appears in the Forms List on the handheld, and how subform records appear in the parent form.

Display as Subform

On the handheld, subforms are not identified as special types of forms. However, to create a subform record, the handheld user has to go into the parent form and choose to add a new subform record, in order for information to be copied from parent to subform. Therefore, the handheld user should never create new subform records by entering the subform directly.

On the handheld, the user can usually create a new record by tapping the name of a form and then tapping the New button. However, if you set the Advanced Form Property of Display as Subform, the user will not be able to create new records by tapping the New button.

In the Forms List, as shown in Figure 8-12, a form that has the Display as Subform property set has a different icon from that of regular forms. The icon identifies that the user cannot create new records directly by tapping the New

button. Instead, the user has to enter the parent form and create a new record from there.

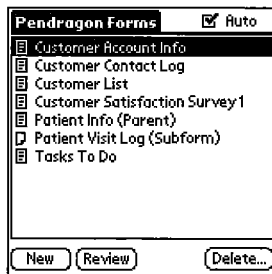


Figure 8-12: A subform that is set to Display as Subform has a different icon than regular forms.

It is still possible to review existing records of the subform directly, without needing to access the parent form.

To set the Display as Subform property, view the Advanced Form Properties window of the subform, and check the check box Display as Subform.

Hide Form in Form List

Instead of displaying a subform as a unique type of form in which the user cannot create records directly, an alternative is simply to hide the subform altogether. By hiding the subform, you can guarantee that the handheld user creates records only from the parent form.

To prevent a subform from being displayed in the Forms List on the handheld, check the Hide Form in Form List check box in the Advanced Form Properties window.



When designing a parent and subform, you may prefer not to hide the subform initially, so that you can verify that the subform has been sent to the handheld. Once the parent and subform are working to your satisfaction, select Hide Form in Form List, and redistribute the subform to the handheld to hide it.

Don't Sort on Subform Screen

When you review a parent record and tap in the subform field, the list of subform records will be sorted according to the Display Key field of the subform.

Sorting the records by Display Key takes time, and if there are a large number of records, you can reduce the length of time it takes to display the list by switching off the sorting feature.

In the Advanced Form Properties window, check the box Don't Sort on Subform Screen.



This property also causes records of a reference form not to be sorted when the handheld user performs a lookup to another form. (See Chapter 5, "Field Types," for information on performing a lookup to another form.)

Automating the Creation of a New Record

Two Advanced Form Properties can be used to speed up data entry by automatically creating a new record.

Auto-Repeat This Form

If you select to Auto-Repeat a form, then when the handheld user exits a record, a new record is automatically created.

Auto-Repeat is useful if you are performing a lot of data entry with the same form, and you want to save a few stylus taps per record by having a new record automatically generate.

To switch on this feature, check the Auto-Repeat check box in the Advanced Form Properties window.

To break the Auto-Repeat cycle, you can tap the End button on a blank record. A dialog box will ask whether you want to keep the record, and you can choose to delete the record. Alternatively, you can tap the handheld Menu button, tap the Record menu, and select Delete/Cancel. (To prevent accidental deletion of records, there is no Graffiti shortcut for deletion.)

Auto-Execute

If you select the Auto-Execute option, then when you tap the Forms icon on the handheld Applications screen, the Forms program will run, you will immediately enter the selected form, and a new record will be created.

Auto-Execute is useful if you have only one form on the handheld, and you want to initiate a new record with the touch of one button.

Only one form in the Forms List can be set to Auto-Execute. If you set this property on more than one form, it will apply only to the first form in the Forms List with Auto-Execute switched on.

Typically, you should also set the Advanced Form Property of Auto-Repeat, so that when one record is complete you can immediately begin a new record for the same form.

To switch on Auto-Execute, check the Auto-Execute button in the Advanced Form Properties window.

If you are in the middle of a record and then switch to another application – for example, the Date Book – tapping the Forms icon will return you to the field in the record that you were previously in.

To exit a blank record, tap the End button and, when prompted, choose to delete the blank record.



You can preprogram one of the hardware buttons on the Palm organizer to run Pendragon Forms when pressed. If the handheld is powered off, pressing a hardware button also turns on the device. On the handheld Applications screen, tap the Prefs icon, and select Buttons. Select the handheld button you want to use to run Forms.

Controlling Deletion on the Handheld

In Pendragon Forms, the default is that you can choose to delete forms and records from the handheld.

Deleting forms from the handheld is not absolute, because it is the User Groups on the PC that determine whether you will receive a form design when you synchronize. Therefore, if you accidentally delete a form from the handheld, the form design will be restored the next time you perform a HotSync data transfer, assuming that the form still exists on the PC and is still assigned to your handheld.



See Chapter 7, "Synchronization Rules," for information on User Groups and deleting forms.

Deleting a form design also deletes the corresponding data. If you want to minimize accidental deletion of form designs, you can use a couple of Advanced Form Properties.



Disabling form deletion and the deletion of records on the handheld protects you from deletion within the Forms application only. It is still possible to delete the entire Forms application, including all forms and data, by going to the Applications screen, tapping the Menu button, and selecting the Delete option.

Disable Form Deletion

If you select to Disable Form Deletion, the handheld user will not be able to use the Delete button in Pendragon Forms to delete a form design from the handheld.

To disable form deletion for a specific form, check the Disable Form Deletion check box in the Advanced Form Properties window.

If you also want to prevent the deletion of records on the handheld, set the access right of No Deletions on PalmPilot, on the Form Properties screen. Select a form and click the Properties button in the Pendragon Forms Manager to access the Form Properties window. See Chapter 3, “Entering Data on the Palm Organizer,” for information on setting access rights on the handheld.



If you disable form deletion for a form design that has been sent to the handheld, do not delete the form from the PC, or you will be unable to remove the form from the handheld unless you delete the entire Pendragon Forms application from the handheld.

Hide Delete Button

The Forms List on the handheld contains a Delete button, as shown in Figure 8-13. On a form-by-form basis, you can select to hide the Delete button, as shown in Figure 8-14.

To hide the Delete button when a user selects a form, check the Hide Delete Button check box in the Advanced Form Properties window.

Hiding the Delete button for a form provides only limited protection for the records in a form. Although the handheld user cannot access the Delete screen to delete the records, it is still possible to delete a record by reviewing the record and selecting a menu option to delete the record.

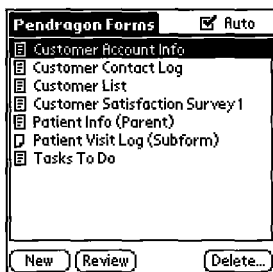


Figure 8-13: The Forms List contains a Delete button.

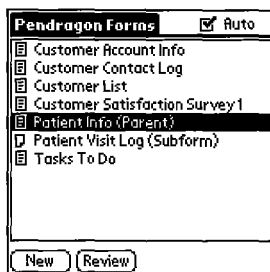


Figure 8-14: You can choose to hide the Delete button.

To prevent any deletion of records from within the Forms application, set the access right of No Deletions on PalmPilot check box in the Form Properties window.

Note that if you hide the Delete button for one form but not for another, it is possible to access the Delete button from the second form, and from the Delete screen it is possible to delete the first form design. To prevent the deletion of a form design, set the Advanced Form Property of Disable Form Deletion.

Advanced Synchronization Rules

As mentioned in Chapter 7, “Synchronization Rules,” several Advanced Form Properties affect synchronization. These Advanced Form Properties work in addition to the Data Persistence options that are set in the Form Properties window to limit the number of records on the handheld.

Allow Inserts on Handheld to Update Existing Records

In Pendragon Forms, the default is that during synchronization, newly created records on the handheld cannot have a Primary Key that matches an existing record in the database on the PC.

The Primary Key is a field or combination of fields that uniquely identify each record. (See Chapter 6, “Advanced Field Properties,” for information on primary keys.)

If the Primary Key is a customer account number, for example, and a newly created record on the handheld is given an account number that matches an existing customer account number, the default synchronization rules would not allow the new record to overwrite the existing record. That way, if the account number had been assigned to the new record by accident, there would be no loss of the old

record after synchronization. The error can then be corrected on the handheld, and the modified new record can be sent to the PC on the next synchronization.

In very rare circumstances, you may want to change this default synchronization rule. For example, say you maintain a database of library books that can be checked in or out, and you are using a bar code number as the primary key. Instead of maintaining a list of all possible books on the handheld, you can use the handheld to create a new record when you scan a book. In the record, you can select whether the item is being checked in or out. Under these circumstances, when you perform a HotSync data transfer, you actually want the new record on the handheld to overwrite the existing record in the database, to update the check-in/check-out status of the book.

To allow new records on the handheld to overwrite existing records in the database, check the Allow Inserts on Handheld to Update Existing Records box in the Advanced Form Properties window.



If the handheld user is manually entering data on the handheld, as opposed to scanning a bar code, there is a risk that if this Advanced Form Property is switched on, an accidental entry on the handheld could overwrite a valid existing record on the PC.

Storage Limit

The Storage Limit is the absolute size limit allowed for storing records of a form on the handheld.

The default Storage Limit is 500 kilobytes per form.

Because the PC can store significantly more records than the handheld, the Storage Limit is designed to offer some protection so that during synchronization, the PC does not try to overflow the memory on the handheld.

The Storage Limit works in addition to the Data Persistence option that you select in the Form Properties window to limit the number of records on the handheld. For example, if you have selected Keep a Copy of Records on PalmPilot, all records will be kept on the handheld up to a maximum of 500 KB.

If you have a form with a lot of records and you have available free space on the handheld, you can choose to increase the Storage Limit, for example to 700 or 1,000KB or more if available.

Alternatively, you can choose to set a smaller Storage Limit.

If there are more records on the PC than can be sent to the handheld, records from the PC will be placed on the handheld in ascending order of Primary Key. The default primary key used by Pendragon Forms includes the creation date and time of a record, and this means that the earliest records will be placed on the handheld first. You can use the Advanced Form Property of Archive Date to set the starting date for placing records on the handheld.

Archive Date

The Archive Date is a cut-off date for placing records on the handheld. Records created before the Archive Date are not placed on the handheld.

The Archive Date relies on the TimeStamp field that is automatically generated on the handheld. The TimeStamp is the creation date and time of a record.

If there are more records on the PC than can fit on the handheld, records from the PC are sent to the handheld in ascending order of the Primary Key. The default Primary Key uses the TimeStamp, so that the earliest created records are sent to the handheld first. If you have months or years of data on the PC, you may not want to place the earliest-created records on the handheld. By setting an Archive Date, you can choose the starting date for placing records on the handheld.

To set an Archive Date, type a date in the short date format used in Windows. For example, your short date format might be mm/dd/yyyy. To check what your short date format is, click Start → Settings → Control Panel → Regional Settings in Windows.



If you are linking to an external Access database or ODBC database, you may not be able to use the Archive Date property. Refer to the next topic, "Additional Download Criteria."

Additional Download Criteria

The Additional Download Criteria option in the Advanced Form Properties window enables you to select which records on the PC are sent to the handheld.

In general, the selections that you make in the Data Persistence section of the Form Properties window are sufficient for determining which records are sent to the handheld.

However, if you are linking to an external Access database or ODBC database, you may not be storing the UserName and TimeStamp fields that are generated on the handheld.

- ◆ The UserName is the handheld user name, and the default synchronization rule is that records with a given UserName are sent to the handheld with the same user name.
- ◆ The TimeStamp is the creation date and time of a record. If the Data Persistence option Keep New Records on PalmPilot for X Days is selected in the Form Properties window, the TimeStamp is used to determine the age of a record. The TimeStamp is also used by the Archive Date property, to determine if a record should be removed from the handheld because the TimeStamp is before the Archive Date.

If you are not storing the `UserName` and `TimeStamp`, you will need to specify Additional Download Criteria in order to select which database records are sent to which handheld.

Even if you are storing the `UserName` and `TimeStamp`, there may be other circumstances under which you may want to use Additional Download Criteria to fine-tune which records go to the handheld.

The Additional Download Criteria is a SQL `WHERE` clause (without the word “where”) that is used to specify the selection criteria for sending records to the handheld. See sidebar topic “SQL (Structured Query Language)”.

The `WHERE` clause that is entered in the Additional Download Criteria box typically has the format:

```
[Database-Column-Name] = Criteria
```

(The “=” sign is one of several relational operators that can be used. Others include greater than `>`, less than `<`, not equal `<>`, `AND`, `OR`.)

When a form is frozen, each field is assigned a database column name. The database column name for a field can be viewed in the Advanced Field Properties window (see Chapter 6, “Advanced Field Properties”). The criteria in the `WHERE` clause is the condition or conditions that must be satisfied for a record to be sent to the handheld.

If you have the full version of Microsoft Access, you can refer to the Access Help on `WHERE` clauses for additional information on the types of selection criteria that you can specify.

SQL (Structured Query Language)

Most modern relational databases, including Microsoft Access, understand a language called SQL. SQL can be used to add, update, delete, or request data from one or more database tables. When a SQL statement is used to request a specific set of records, a “WHERE” clause is added at the end of the statement to specify which records will be affected by the statement.

When you change the synchronization rules for a form, Pendragon Forms automatically generates a SQL statement which requests the appropriate records from the database. For example, setting a Data Persistence option that will keep records on the handheld for 5 days, generates an SQL statement that includes a `WHERE` clause such as:

```
WHERE TimeStamp > NOW - 5
```

To give your synchronization rules more flexibility, Pendragon Forms enables you to add your own criteria to the `WHERE` clause that Pendragon Forms generates. The text of the criteria that you enter into the Additional Download Criteria box is simply inserted into the SQL statement as part of the existing `WHERE` clause.

DEFAULT SETTING: EACH HANDHELD IS ASSIGNED SEPARATE RECORDS

The default Additional Download Criteria setting is:

```
[UserName] = ##USERNAME##
```

##USERNAME## is a wildcard used by Pendragon Forms to represent the user name of the handheld that is currently synchronizing.

The default setting means that the only records sent to the handheld are those in which the UserName field matches the handheld user name. This setting makes it possible for you to assign different records to different users.

SHARING RECORDS ACROSS HANDHELD DEVICES

If you want to create an application in which several handheld users each share the same record, you need to do two things:

1. **Select your own Primary Key for the form.** By default, Pendragon Forms uses the UserName and TimeStamp as part of the Primary Key. When a handheld user modifies a record, that person's UserName is assigned to the UserName field. If the same record is sent to two handheld devices, and one user updates a record, a duplicate record will be created on the PC when the user synchronizes. This is because the Primary Key on the handheld will differ from the Primary Key on the PC, because of the changed UserName.

If you do not want to create duplicate records when handheld users are sharing records, select your own Primary Key instead of using the default Primary Key. See Chapter 6, "Advanced Field Properties," for more information on primary keys.

2. **Delete the default Additional Download Criteria.** To allow records with any UserName to be sent to every handheld, delete the default Additional Download Criteria setting of [UserName] = ##USERNAME## in the Advanced Form Properties window.



If you choose to allow multiple handheld devices to share the same record, there is a danger that if two users update the same record and then perform a HotSync data transfer, the changes made by the first person to synchronize will be overwritten by the last person to synchronize. Refer to Chapter 15, "Planning a Multi-User Installation," for more information on the issues involved when sharing records.

DATE CRITERIA

If you select the Data Persistence option to Keep New Records on PalmPilot for X days, Pendragon Forms uses the creation date and time of a record, stored in the TimeStamp field, to determine when to remove a record from the handheld.

If you are linking to an external Access or ODBC database and you are not storing the TimeStamp field, you can use your own date criteria based on a date field of your choice in the Additional Download Criteria to determine when to remove records from the handheld.

Even if you are using the TimeStamp field, you may want to create date criteria that depend on a field that is not the creation date of the record.

For example, your form may contain a field called Project Ending Date, and you may want to keep a record of projects 30 days after their completion. You can add the following criteria in the Additional Download Criteria field:

```
[ProjectEndingDate] > now - 30
```

The Project Ending Date has to be a Date or Date & Time field on your form. The word `now` is a function in Microsoft Access SQL that returns the current date and time. In the previous example, `now - 30` means thirty days prior to the current date and time.

Microsoft Access stores dates internally as numbers. `[ProjectEndingDate] > now - 30` means that the ending date of the project must come after a date which is thirty days prior to the present date. If the Project Ending Date is thirty-one days ago, the internally stored number will be less than thirty days from the present, and so the record will not be sent to the handheld.

If you have, for example, a Date of Visit field on your form, and you want to send all records after September 30, 1999, to the handheld, you can enter the following Additional Download Criteria:

```
[Date of Visit] > #09/30/1999#
```

Type the date in the Windows short date format used on your PC. The `#` symbol is used by Microsoft Access as a quotation symbol when you are specifying dates in an SQL statement.

SELECTION LIST CRITERIA

You can determine which records to send to the handheld by using criteria that depend on the response in a Yes/No field, a Popup List, or a Lookup List. Selection fields are preferred over Text fields, because you cannot guarantee that a Text field will contain a specific response, whereas you know exactly what the items in a selection field will be.

If you have a Popup List that allows the user to select whether the status of a Work Order is New, In Progress, or Complete, you can set up Additional Download

Criteria to send only New or In Progress records to the handheld. The WHERE statement is:

```
([WorkOrderStatus] = "New") OR ([WorkOrderStatus] = "In Progress")
```

The OR operator allows for either the first criterion (New) or the second criterion (In Progress) to apply.

With Selection fields, it is also a good idea to take into account the case in which the field has been left blank. If you want to send records to the handheld if the field is blank, you can use the following WHERE statement:

```
([WorkOrderStatus] = "New") OR ([WorkOrderStatus] = "In Progress")  
OR ([WorkOrderStatus] is null)
```

COMPLETION CHECKBOX CRITERIA

When using a Completion Checkbox field, you can use the Data Persistence option Keep Incomplete Records on PalmPilot to remove completed records from the handheld. Data Persistence options are set in the Form Properties window.

If you are linking to an external Access or ODBC database, and your Completion Checkbox field maps to a Text field in the external database, then you can use the Data Persistence option to manage the removal of records from the handheld.

However, if your Completion Checkbox field maps to a Yes/No (Boolean) field in your database, the Data Persistence option will not work. This is because the Completion Checkbox field in Pendragon Forms is actually internally stored as a Text field with the possible values Y, N, or null.

To obtain the same effect as the Data Persistence option of Keep Incomplete Records on PalmPilot when using a Yes/No field in the external database, you can create your own Additional Download Criteria similar to the following:

```
([RepairTechnician] = ##USERNAME##) AND ([CompletedWork] is null) OR  
[CompletedWork] = 0)
```

In this example, the Repair Technician field in the external database records the handheld user name, and the Completed Work field stores the Completion Checkbox field.

SORTING RECORDS AS THEY DOWNLOAD TO THE HANDHELD

Based on your form definition and download criteria, Pendragon Forms creates a database query to download the appropriate records to the handheld. By default, the query returns records sorted on the primary key columns, though this may vary depending on the database version you are using to store the data.

You can control the sort order of records sent to the handheld by adding a SQL ORDER BY clause to the Additional Download Criteria.

In structured query language (SQL), the ORDER BY clause looks like the following:

```
ORDER BY column1, column2, column3 DESC
```

There can be one or more column names, and each may be followed by the keyword DESC. The DESC keyword causes the column to be sorted in descending instead of ascending order.

For example, if you want records restricted by user name (the default) but also want records sorted by a Date of Visit field, your Additional Download Criteria will look like this:

```
UserName = ##USERNAME## ORDER BY [DateOfVisit]
```

If you want the same data sorted in descending order (most recent dates first), add the word DESC to the end of the criterion.

Note that some columns cannot be sorted in this way. Signature fields and text fields longer than 255 characters cannot be sorted, and an error message will be entered into the HotSync log when you synchronize the form.

Summary

Advanced Form Properties can hide buttons on the handheld, control the deletion of form designs, and give the handheld user feedback on missing Required fields. Advanced Form Properties can also be used to cause a form to Auto-Repeat, so that as soon as the user ends one record you begin a new record. Finally, advanced synchronization options can be specified via the Advanced Form Properties screen.

Part

Managing Data and Form Designs

IN THIS PART

CHAPTER 9

Managing Data on the PC

CHAPTER 10

Managing Form Designs

Chapter 9

Managing Data on the PC

IN THIS CHAPTER

- ◆ Viewing and editing data in the database
- ◆ Exporting data to Excel or to ASCII
- ◆ Creating reports
- ◆ Backing up the Pendragon Forms database
- ◆ Backing up individual form designs and data

PENDRAGON FORMS SUPPORTS bi-directional synchronization. This means that you can choose whether you want to create new records on the handheld to upload to the PC, or whether you want to do the reverse: create records on the PC to send to the handheld. You can also do a combination of the two if it suits you.

In Pendragon Forms, the PC is the central storage place for both form designs and data. Once data has been sent to the PC during a HotSync data transfer, the data can be edited or exported, or you can create reports.

Where Is the Data from the Handheld Stored?

The Pendragon Forms Manager is a Microsoft Access 97 database (or, if you are using Microsoft Access 2000, Pendragon Forms Manager is an Access 2000 database). All form designs and data are stored in a single database file.

- ◆ The default name of the database file is Forms3.mdb (or Forms32k.mdb if you are using Access 2000).
- ◆ The location of the database file is typically the C:\Program Files\Forms3 folder.

Within the Forms3.mdb file (or Forms32k.mdb file), there is a separate database table for each form that you have frozen. Data for a given form design is sent to the appropriate database table during the HotSync process.



To find out the database table name for a form, click the name of the form in the Forms Manager and then click the Properties button. The Table Name field lists the database table name for that form. All database table names associated with form designs begin with FORM_ID_.

Viewing and Editing Data in the Database

The Pendragon Forms Manager contains some basic tools for viewing and editing data in the database. The Data Functions section of the Forms Manager, shown in Figure 9-1, contains buttons for accessing the data in a form.

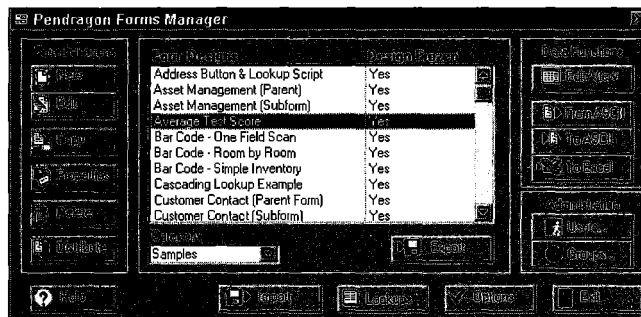


Figure 9-1: The Pendragon Forms Manager

Datasheet View

To view the records in a form, click the name of the form and then click the Edit/View button. A form must be frozen before you can access it with the Edit/View button.

The default view of a form is an Access form in datasheet view. As shown in Figure 9-2, datasheet view displays the fields on the form as columns, and each record as a separate row.

If a column is too narrow for your liking, you can position the cursor in the gray cell above the column name, on the border between two cells. The cursor will change shape to a crosshairs icon (+), and you can click and drag the border to make the column wider. The same applies to making rows taller, except that with rows you need to click in the gray cell to the right of a row.

RecordID	UnitID	UserName	TimeStamp	CustomerName	DateOfPurchase	TypeOfHar
0	0	Debra Sanch	06/11/1999 1:29:38 PM	Belinda	06/11/1999	Palm VII
0	0	Debra Sanch	06/13/1999 12:10:18 PM	Filbert Jones	06/13/1999	Workpad
0	0	Debra Sanch	06/13/1999 12:19:46 PM	Clarinda	06/13/1999	Palm IIIx
0	0	Debra Sanch	06/14/1999 8:55:04 PM	Evan	06/14/1999	Palm VII
0	0	Debra Sanch	06/15/1999 7:27:50 PM	Zulre	06/15/1999	Palm VII
0	0	Debra Sanch	06/15/1999 11:36:15 PM	P. McOtter	09/30/1999	Palm IIIx
0	0	No one	06/17/1999 7:45:12 PM			

Figure 9-2: Viewing the data in the form in datasheet view

The column names are derived from the names of the fields on the form. Each column name comprises the first 64 characters of the corresponding field name. Spaces and punctuation are removed from the field name to generate the column name.

In every form, you will see four columns in addition to the columns for the fields on your form. These columns are: RecordID, UnitID, UserName, and TimeStamp. Pendragon Forms generates these four fields each time a record is created. Three of these fields, UnitID, UserName, and TimeStamp, are used as the default primary key to uniquely identify individual records.

- ◆ RecordID is generated on the handheld. Each record on the handheld has a unique RecordID number. Because this information is used only on the handheld, it is not uploaded to the PC, and when you view data, the RecordID column will be blank. This column is maintained primarily for backward compatibility with earlier versions of Pendragon Forms.
- ◆ UnitID defaults to zero. If records are imported into the database, and the TimeStamp field is not unique, the UnitID field can be used to force individual records to be unique by assigning a unique number to this field.
- ◆ UserName is the handheld user name. The default synchronization setting is that only records with a given UserName are sent to the handheld device with the matching name.
- ◆ TimeStamp is the creation date and time of a record. The TimeStamp is used to determine when records should be removed from the handheld.

Editing a Record

If you need to make corrections to a record, click in a cell (that is, a field in a row) and type your corrections. To save your changes, position the cursor outside of the row. You will notice that the gray cell to the left of a row will contain an icon of a pencil while a record is being edited and will contain a solid triangle after the cursor has exited the row. The solid triangle indicates that the changes have been saved.

If you are editing Popup Lists and Lookup Lists, clicking in a cell will display the items in the list, as shown in Figure 9-3. The exception is a Cascading Lookup field – datasheet view is not able to display the Lookup List in this case.

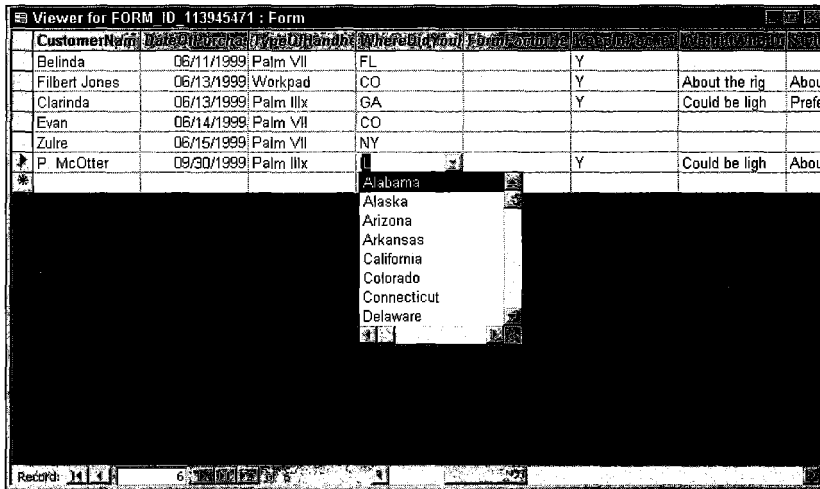


Figure 9-3: Clicking in a Popup List or Lookup List field displays the items in the list.

Adding a New Record

You can add new records on the PC, and the records will be sent to the handheld during the next HotSync data transfer.

To add a new record:

1. Click in the last row in the list of records. (A new blank row will be created below the row in which you clicked.)
2. Leave the RecordID and UnitID fields blank.
3. In the UserName field, select the handheld user who is to receive the record. If you are sharing records across handhelds (see Chapter 15, "Planning a Multi-User Installation"), you can leave the UserName as the default of No One.

4. The TimeStamp field is automatically generated.
5. Position the cursor in a cell and type or select the data that you want to enter in the record. Press Tab to move to the next cell.
6. Position the cursor out of a row to save the record. When you close the Edit/View window and perform a HotSync data transfer, the new record will be sent to the specified handheld, assuming that the form design has been distributed to the handheld.

Table 9-1 shows the types of data that you can enter in the different field types. Note that when you are viewing data, there is no obvious way to tell what field type a given field is. If you cannot tell what the field type is from the name of the field, you may want to print a copy of the form design to have nearby when you enter records on the PC. Chapter 2, “Creating a Form,” outlines how to print form designs.

TABLE 9-1 ENTERING DATA ON THE PC

Field Type	Data Format
Text	Type any characters. For a carriage return, press Ctrl+Enter. (If you press Enter only, you will move to the next cell, not create a carriage return.)
Numeric	Type numeric characters only. A decimal point can be used.
Currency	Type a number with two decimal places. If you do not add a decimal place, a whole number of dollars is assumed.
Option 1 of 5	Type one of the following numbers: 1, 2, 3, 4, or 5. You will need to know from your form design which fields are Option 1 of 5 fields.
Popup List, Lookup List, Exclusive Lookup List	Click in the cell to display the list. Select an item from the list.
Multi-Selection List	Type the name of an item in the list, followed by a semicolon (;), followed by the next item that you want to include. Example: Red;Blue;Green. (Don't include the period as shown in the example.) You will need to know the names of the items from your form design. Items that you list will be shown as selected on the handheld.

Continued

TABLE 9-1 ENTERING DATA ON THE PC (Continued)

Field Type	Data Format
Date Only, Date & Time, Time	Date fields should be entered in the short date format used in Windows. Example: 06/30/1999. To see what short date format you are using, click Start → Settings → Control Panel. Double-click the Regional Settings icon and click the Date tab to view your short date format. The Time part of a Date & Time field, or a Time field, should be entered in the Time format used in Windows. Example: 06/30/1999 09:30 P.M. for a Date & Time field, or 11:25 A.M. for a Time field. The time format used in Windows is also accessed from the Regional Settings icon in the Control Panel.
Yes/No Checkbox	Type Y to represent Yes and N to represent No. You will need to know from your form design which fields are Yes/No fields.
Time Checkbox, Completion Checkbox	Generally these field types should be filled in only on the handheld. A Time Checkbox records a date and time that is visible only on the PC – to the handheld user, this field looks like a check box. The Completion Checkbox is used by the handheld user to flag a record for removal from the Palm organizer.
All other field types	Leave blank on the PC. All other field types are not capable of sending data to the handheld. For example, a Jump Popup field or a Button field never contains data.

Deleting a Record

Deleting a record from the PC will remove the record from the handheld only if the record has not been modified on the handheld. If a record has been updated on the handheld, then it will be uploaded to the PC during the next synchronization. The record can be deleted from the PC after the HotSync process.

To delete a record, select the form in the Forms Manager and click the Edit/View button to view the data. Click in the gray cell to the left of the row containing the record. The entire row will be highlighted. Press the Delete key on the keyboard. You will be prompted to confirm the deletion. Click Yes to delete or No to cancel.

Alternative Methods of Viewing Data

Selecting a form and clicking the Edit/View button displays the records in the form in datasheet view, which displays all the records as rows on screen. There are three alternative ways to view the data on the PC.

- ◆ Press **Ctrl+Edit/View** button. This displays an Access form in Form View, which shows one record at a time. As shown in Figure 9-4, there are arrow buttons at the bottom of the screen to move from one record to the next. The arrow button with a star (*) is used to create a new record.

Figure 9-4: Viewing data in Form View

- ◆ Press **Shift+Edit/View** to display the data in the form of a query. Figure 9-5 shows the records as a query. Unlike datasheet view and form view, Popup Lists and Lookup Lists do not appear as selection lists in query view. The main reason for using query view is if your form is so large that Access is not able to display the form in datasheet view or in form view.

RecordID	Unfill	Usak	Time	TimeStamp	Discretion	Location	Name
0	0	Debra Sancho	399 1:29:38 PM		Belinda	06/11/1999	Palm VII
0	0	Debra Sancho	39 12:10:18 PM		Filbert Jones	06/13/1999	Workpad
0	0	Debra Sancho	39 12:19:46 PM		Clarinda	06/13/1999	Palm IIIx
0	0	Debra Sancho	399 8:55:04 PM		Evan	06/14/1999	Palm VII
0	0	Debra Sancho	399 7:27:50 PM		Zulre	06/15/1999	Palm VII
0	0	Debra Sancho	39 11:36:15 PM		P. McOtter	09/30/1999	Palm IIIx
0	0	No one	399 7:54:37 PM				

Figure 9-5: Viewing data in the form of a query

- ◆ Press Alt+Edit/View to display the actual Access database table containing the data. This method is useful if you want to find out the name of the database table where records for a form are being stored. As with viewing the form as a query, Popup Lists and Lookup Lists are not available as selection lists if you edit the database table.

Creating a Simple Work Order

One of the biggest demands for Pendragon Forms is for use as a work order system. From repair technicians to swimming pool cleaners, the need for work orders crosses all industries.

In a simple work order scenario, work requests come into the office, and the work orders are assigned to individuals who go into the field to complete the work orders. When the work has been completed, the office needs to receive the data in order to generate billing or other report information.

Because Pendragon Forms supports bi-directional synchronization, it is very easy to create a work order system in which the field workers use handheld devices to receive their work instructions and to log the work performed.

Following are some of the features of Pendragon Forms that are used to create a work order system:

- ◆ Records can be created on the PC. The person in the office receiving the work request from the customer can create a record in the Pendragon Forms Manager. The record can be partially filled on the PC and then sent to the handheld for completion in the field.

- ◆ By selecting the appropriate handheld UserName when the record is being created, you can assign work orders to different handheld devices. In this way, each field worker can have a set of tasks to perform or customers to visit.
- ◆ If a Completion Checkbox is used on a form, it enables the field worker to control when a record is removed from the handheld. A task that is not completed can remain on the handheld until the user checks the Completion Checkbox to flag the record as complete.



If you installed Pendragon Forms from the CD-ROM, open the Pendragon Forms Manager and select the Samples category. The Work Order Dispatch form is a simple work order form. On the PC, create one or two new records for the form, selecting your handheld user name in the UserName field. Fill in only the first few fields of the form — the remainder can be filled in on the handheld. Distribute the form to your handheld, and your assigned records will also be sent to the handheld during the next HotSync data transfer.

In the sample Work Order Dispatch form on the CD-ROM, the person receiving the work request from the customer can create a new record on the PC, as shown in Figure 9-6. Only the fields relative to the appointment date, the customer, and the task to be performed are filled in on the PC. You assign the work order to a worker by selecting that person's handheld user name in the UserName field.

RecordID	0
UnitID	0
UserName	Debra Sancho
TimeStamp	06/16/1998 8:30:02 AM
Appt Date_Time	06/16/1998 9:00:00 AM
Cust. Last Name	Marvin
Cust. First Name	Cassie
Cust. Address	124 E. Walnut Trail
Type of work to be perform	Installation
Description of work	Install new washer and gas dr
Start Time	
End Time	
Hours Worked	
Rate per hour	
Total Bill	
Signature	Open...

Figure 9-6: A new work order record is created on the PC.

When the handheld user performs a HotSync data transfer, the new work order is sent to the handheld. To access the work orders, the handheld user taps the name of the form and taps the Review button to view the list of outstanding work orders.

As Figure 9-7 shows, when the handheld user selects a work order to review, the fields that were filled in on the PC provide information on the work to be performed.

6/16/99 9:00 am	
Appt Date & Time	6/16/99 9:00 am
Cust. Last Name	Marvin
Cust. First Name	Cassie
Cust. Address	124 E. Walnut Trail
Type of work to be	Installation
Description of wor	Install new washer
Start Time	<input type="checkbox"/>
Resolution	
End Time	<input type="checkbox"/>
Hours Worked	
Rate per hour	\$0.00
<input type="button" value="End"/> <input type="button" value="Up"/> <input type="button" value="Left"/> <input type="button" value="Right"/> <input type="button" value="Down"/>	

Figure 9-7: The work order on the handheld

As the work is performed, the handheld user fills in the blank fields on the form to create a record of the work done. Checking a Completion Checkbox, as shown in Figure 9-8, marks the record for removal from the handheld on the next HotSync data transfer.

6/16/99 9:00 am	
Cust. First Name	Cassie
Cust. Address	124 E. Walnut Trail
Type of work to be	Installation
Description of wor	Install new washer
Start Time	<input checked="" type="checkbox"/>
Resolution	Washer and dryer i
End Time	<input checked="" type="checkbox"/>
Hours Worked	1.25
Rate per hour	\$25.00
Total Bill	\$31.25
Complete?	<input checked="" type="checkbox"/>
<input type="button" value="End"/> <input type="button" value="Up"/> <input type="button" value="Left"/> <input type="button" value="Right"/> <input type="button" value="Down"/>	

Figure 9-8: The completed work order is marked for removal from the handheld.

When the handheld user synchronizes at the end of the day, the record is updated on the PC, as shown in Figure 9-9. (If the form contains a Signature field, the signature is visible if you use Ctrl+Edit/View to view records that are uploaded from the handheld.) The details of the work performed can then be used to generate a report.

RecordID	0
UserID	0
User Name	Debra Sanchez
Time/Date	06/16/1999 8:30:02 AM
App Date/Time	06/16/1999 9:00:00 AM
Est. Last Name	Marvin
Est. First Name	Cassie
Est. Address	124 E. Walnut Trail
Type of Work to be performed	Installation
Description of work	Install new washer and gas d
Start Time	06/17/1999 9:00:10 AM
Final Note	Washer and dryer installed and
End Time	06/17/1999 10:15:12 AM
Hours Worked	1.25
Hourly Rate	\$25.00
Total Bill	\$31.25
Signature	<i>Cassie Mann</i>
Completed?	Y

Figure 9-9: The record on the PC is updated after the handheld user synchronizes.

Creating a Report in Microsoft Access

If you have the full version of Microsoft Access 97 or later, you can use Access to create a report based on data in a Pendragon form. (This topic goes beyond the scope of the Pendragon Forms program itself and covers Microsoft Access features that can be used with data from Pendragon Forms database.)



If you do not have the full version of Microsoft Access 97 or later, this report option is not available.

In the Pendragon Forms Manager, click the name of a form to select the form. Bring the Forms3: Database window to the foreground in order to create a report in Microsoft Access. If you do not see the Forms3: Database window, click the Windows menu and choose Cascade.

Figure 9-10 shows the Forms3: Database window in the foreground and the normal Pendragon Forms Manager window in the background. The Forms3: Database window is a normal Access database window with all of the features of Access.

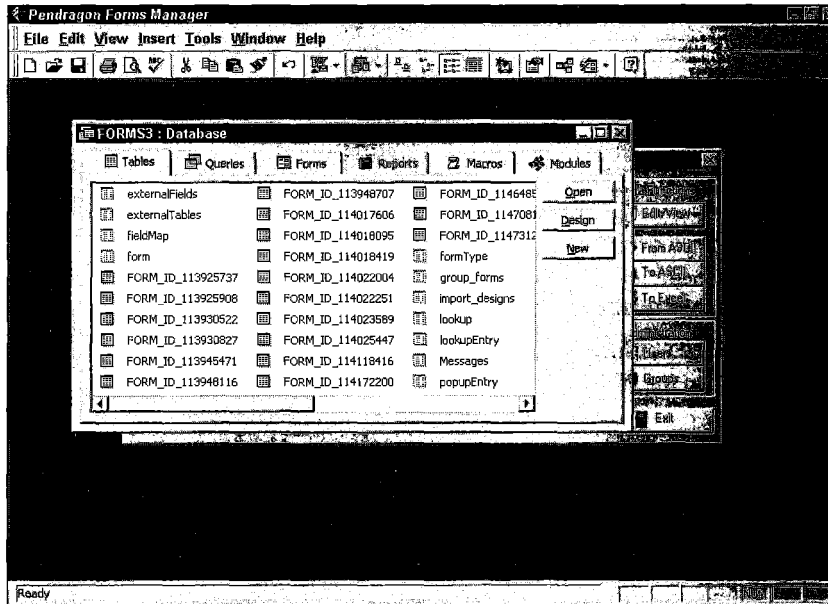


Figure 9-10: The Forms3: Database window

Pendragon Forms has a report generator tool to help you create a report. Click the Forms tab in the Forms3: Database window and then double-click the Report Generator form in the list of forms. A Report Generator window appears, as shown in Figure 9-11.

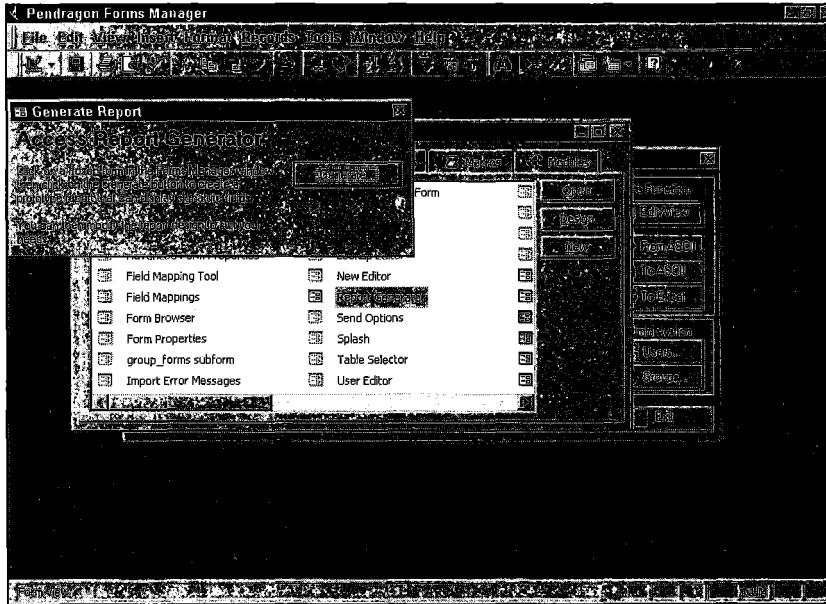


Figure 9-11: The Report Generator window

Click the Generate button to generate an Access report in design mode. Figure 9-12 shows the report in Design View mode.

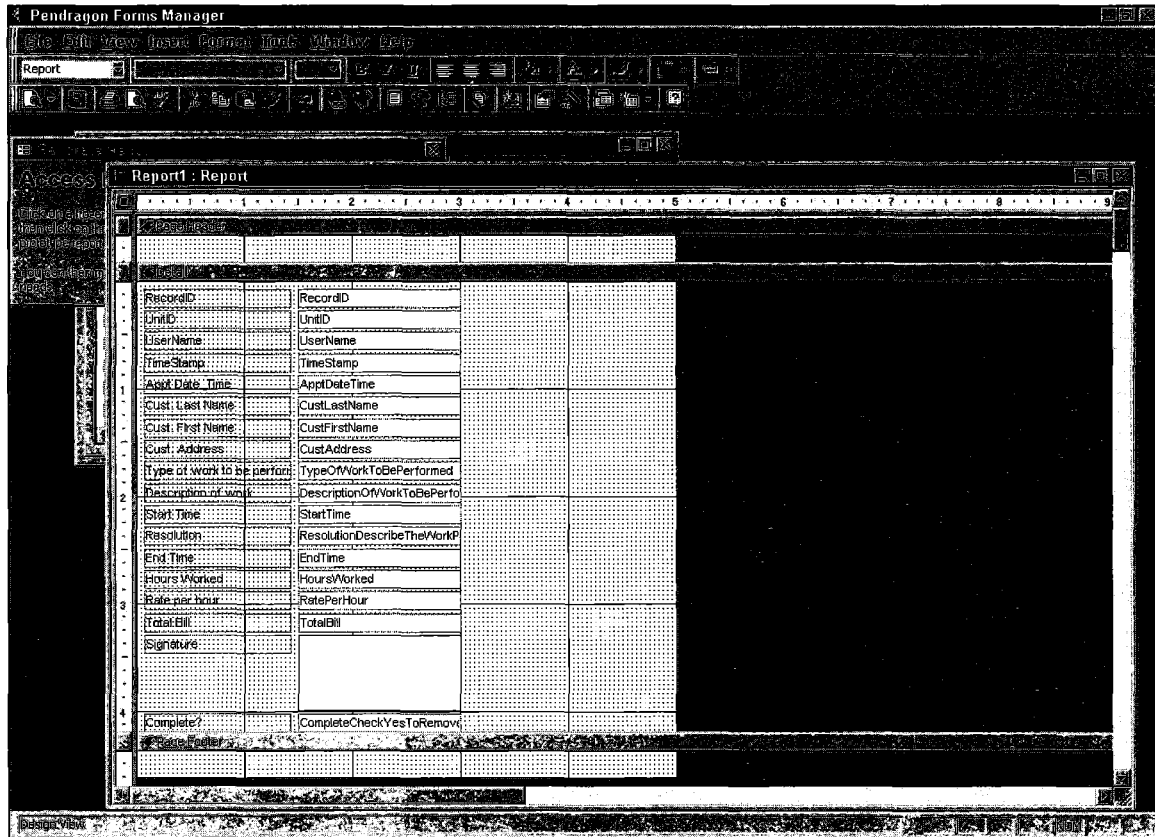


Figure 9-12: The Access Report in Design View mode

Click the View button to view what the report will look like. Figure 9-13 shows the report in Print Preview mode.

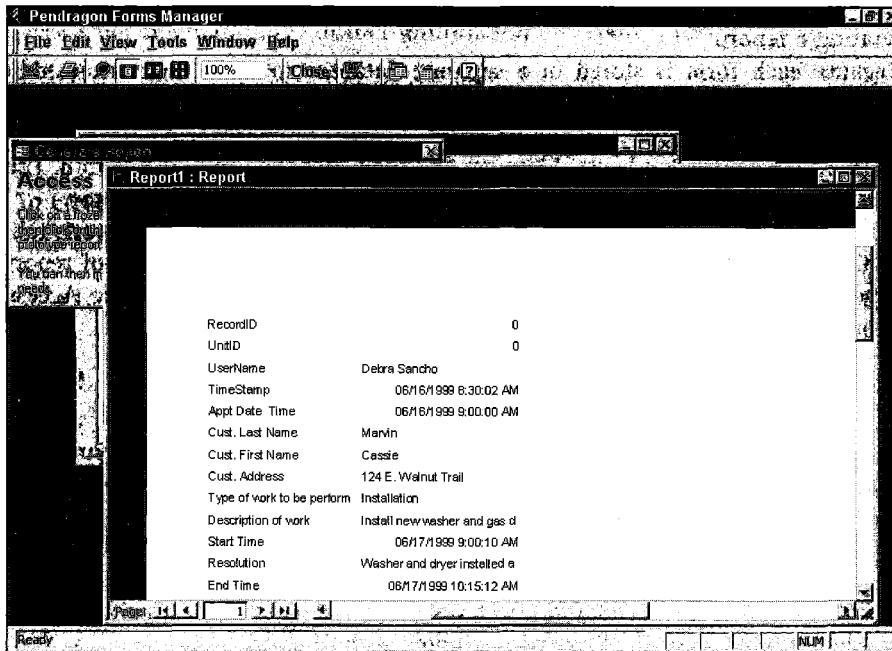


Figure 9-13: The Access Report in Print Preview mode

To modify the report, click the View button again to go back to Design mode. (Note that the View button is a toggle button that uses two icons – when you are in Design mode, the View button looks like a magnifying glass over a sheet of paper, and clicking the button switches you to Print Preview mode; when you are in Print Preview mode, the View button looks like a triangle with a pencil, and clicking the button switches you to Design view.) Refer to your Microsoft Access documentation for information on designing reports.

The main advantage of using the Pendragon Forms Report Generator is that it has a built-in capability to display signatures on a report. If your form has a Signature field (see Chapter 5, "Field Types"), you can use the Pendragon Forms Report Generator as the basis for the design of your form, and then modify the design to suit your needs.

An Alternative Method for Creating an Access Report

If your form does not have signatures, you can use the normal Access method of generating a report.

Because each form is stored in a separate database table in the Forms3.mdb database (or Forms32k.mdb if you have Access 2000), you will need to identify which database table corresponds to your form design. To do this, click the name of the form in the Pendragon Forms Manager and then click the Properties button. The Table Name field in the Properties window identifies the database table name.

Bring the Forms3: Database window to the foreground and click the Reports tab. Click the New button to display the New Report window. Select the type of report that you want to create, and select the database table name. Follow the Access prompts to create the report.

Adding a Filter to a Report

Microsoft Access enables you to create a filter for a report, so that when you generate a report, you do not print every record every time. A useful filter is a date filter, which enables you to print reports for a specific date.

To add a filter to your Access report, you need to be in Design View mode in the report. Click View → Properties. The Properties window is displayed. Enter your filter criteria in the Filter field and select Yes in the Filter On field to switch on the filter. Figure 9-14 shows the Properties window with filter criteria.

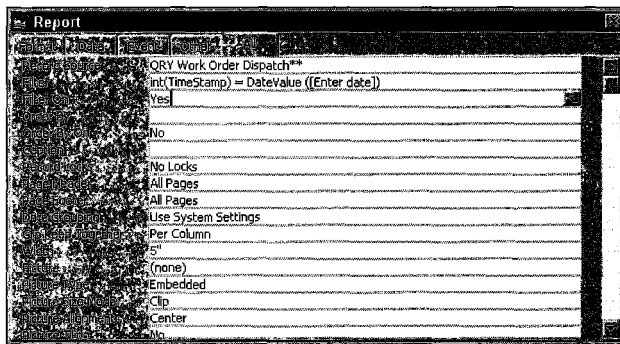


Figure 9-14: A report with filter criteria

To create a report that prints all records created on a particular day, enter the following code to the Filter field:

```
int(TimeStamp) = DateValue([Enter date])
```

When you switch to Print Preview mode, the report will prompt you to enter a date, and only records created on that date will have reports generated.

You can create many different types of report filters. It is beyond the scope of this book to detail how Microsoft Access works, so you may want to refer to your Access documentation for further information on report filters.

Creating a Report in Microsoft Excel

To create a report in Microsoft Excel, you can export data from Pendragon Forms to Excel.

1. In the Pendragon Forms Manager, click the name of a form to select the form.
2. Click the To Excel button to export the records in the form to an Excel spreadsheet. You will be prompted for a filename for the Excel spreadsheet.
3. After exporting the data to Excel, you can close the Pendragon Forms Manager and open the Excel file to create your report.

The Excel file contains the RecordID, UnitID, UserName, and TimeStamp columns, which Pendragon Forms uses to uniquely identify individual records. If you do not need these columns in your report, you can delete them.

If you attempt to export data to Excel and get an error message about Installable ISAMs, refer to Appendix A, "Troubleshooting Tips."

Exporting Data to ASCII (CSV) Format

If you have the full version of Microsoft Access, you may want to use Access to export data, because Access offers several options for data formats that can be exported.

If you do not have Microsoft Access, and you want to export data to ASCII, the Pendragon Forms Manager gives you the option to export data to a comma-delimited format (also called comma-separated variable or CSV format).

To export the data in a form, click the name of the form and then click the To ASCII button. You will be prompted to for a filename for the .csv file.

If you attempt to export data to ASCII and get an error message about Installable ISAMs, refer to Appendix A, "Troubleshooting Tips."

Creating a Report in Microsoft Word

If you do not have the full version of Microsoft Access, you can create reports by performing a Mail Merge to Microsoft Word.

In a Mail Merge, two files are needed: the report that is created using Word, and the data file with the records to be merged into the report. To generate the data file, data is exported out of Pendragon Forms in ASCII format.

The following instructions can be used with Microsoft Word 97. If you have a different version of Microsoft Word, consult your Microsoft Word documentation for instructions on performing a Mail Merge with data from a comma-delimited ASCII file.

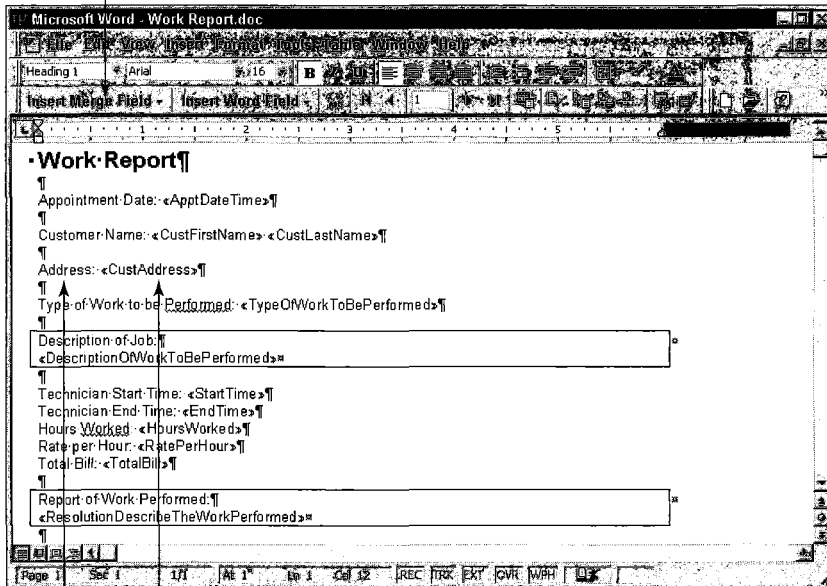
1. In the Pendragon Forms Manager, click the name of a form and then click the To ASCII button.
2. An Export to Text window appears. Select a directory folder and type a name for the Text file to which the data will be exported. Click Save.
3. Close the Pendragon Forms Manager.
4. Open Microsoft Word and choose to create a New, blank document.
5. Click Tools → Mail Merge. The Mail Merge Helper dialog box appears.
6. Click the Create button and choose Form Letters.
7. You will be prompted for a document to use. Choose Active Window.
8. In Step 2 of the Mail Merge Helper, click the Get Data button and choose Open Data Source.
9. In the Open Data Source window, change the directory folder to view the folder where you stored the Text file (in step 2) and select to view files of type *.txt. Double-click your selected file.
10. Word will display a dialog box prompting you to click the Edit Main Document button, as shown in Figure 9-15. Click this button.



Figure 9-15: The dialog box in Word that prompts you to edit your report to add Merge fields

11. You will be returned to the new, blank document. Word will have an additional button, Insert Merge Field, shown in Figure 9-16. The Insert Merge Field button enables you to position fields from your form on the report.

Insert Merge Field button



Field

Text in report

Figure 9-16: The Microsoft Word window with the Insert Merge Field button

12. Type the text of your report. To position a field on your report, click the Insert Merge Field button and select the field of your choice. The field will appear in angled brackets, as shown in Figure 9-16.
13. Save the Word document. This document will serve as your report template and can be used over and over.
14. To merge the data into the report, click Tools → Mail Merge. In Step 3 of the Mail Merge Helper window, click the Merge button.

15. A Merge window appears. Choose to merge to a New document and then click the Merge button. The records from your form will be merged into your report. Figure 9-17 illustrates a finished report.

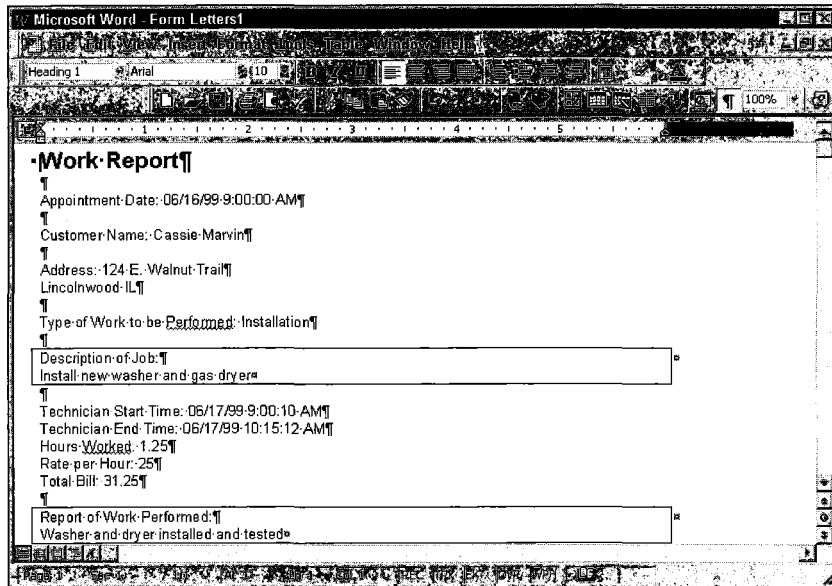


Figure 9-17: Data from the form after being merged into the Word report

Importing Data from an ASCII (CSV) File

If you have data in a comma-delimited ASCII format (also called a comma-separated variable or CSV file), you can import the data into a form on the PC, in order to populate records on the handheld.

Format of a CSV File

A comma-separated variable or CSV file is a text file in which the first row identifies the field names, and all subsequent rows are individual records. Fields within each row are separated by a comma, hence the name “comma-separated variable.” Text is surrounded by double quotes.

Here is what a CSV file looks like:

```
"Name","Address","IDNumber","FavoriteColor"  
"John Smith","44 East End Blvd",22724,"Blue"  
"Paula Sciorra","171 Huntley Creek Rd.",2140,"Green"  
"Smiley Phillips","71 Barley Lane",41039,"Purple"  
"Anna Jones","981 W. 40 St.",6241,"Yellow"
```

Importing a CSV File into a Database Table

For data to be imported correctly from the CSV file into the database table for a given form, the column names in the CSV file must exactly match the column names in the Forms database table.

Create your form in Pendragon Forms and freeze the form design. To find out what the column names in the database table are, you can do two things:

- ◆ Select the form and then click the Edit/View button. Manually enter one test record. Then close the Edit/View window, click the name of the form, and click the To ASCII button to export the data to comma-delimited ASCII. You will be prompted for a name of the ASCII file. After saving the file, open the file in Windows Notepad and look at the column names in the first row of the ASCII file.
- ◆ Alternatively, to view database column names in the Pendragon Forms Manager, you can click the name of a form to select it. Then press Shift+Edit/View. This will display the database table. You may need to widen the columns in order to view the entire column names.

Make a note of the exact column names used in the database table and use the same column names in your CSV file.

To import the CSV file into the database table, click the name of the form in the Pendragon Forms Manager, and then click the From ASCII button. Select the ASCII file to import. You will be prompted to enter a handheld user name to whom the records will be assigned. If you do not want to assign the records to a handheld user at this time, leave the default of No One. (In a single-user scenario, select the handheld user name.)

After the import is complete, click the Edit/View button and view the records to verify that the import proceeded correctly.

Creating a CSV File from Microsoft Excel

If you have data in Microsoft Excel that you want to use to populate records in a Pendragon form, you can choose to save the data in a comma-delimited ASCII format (file type .csv). Once the data is in a CSV format, it can be imported into the database table of a Pendragon form.

As Figure 9-18 illustrates, the Excel spreadsheet must contain column names on the first row and individual records on subsequent rows. The column names must exactly match the column names in the Pendragon Forms Manager database table for the form.

FirstName	LastName	DateofVisit	ColorRequested	Quantity
Millie	Jones	09/10/1999	Blue-Green	5
Fran	Sampson	09/10/1999	Orange	1
Jennie	Valery	09/10/1999	Red	2
Mandy	Jeffers	09/11/1999	Red	3
Julie	Anson	09/11/1999	Blue-Green	1
Helga	Vincent	09/11/1999	Purple	10
Andie	Rich	09/15/1999	Red	3
Sarah	Smyth	09/15/1999	Purple	2

Figure 9-18: An Excel file can be saved as an ASCII CSV file.

Keep your file in its original format – that is, as an Excel worksheet. In addition, however, make a copy of the file and choose to save as file type .csv – comma-delimited ASCII. After you have saved the .csv file, Excel may prompt you to save the file as a worksheet; cancel this option in order to retain the file in CSV format.

Once you have created the .csv file, you can import the data in the file into a Pendragon form and then send the records to the handheld.

Protecting Your Work

Protecting your work is important when using any software application, and working with Pendragon Forms is no exception. By making regular backups of your work, you can increase your chances of recovery in the event of a technical problem such as a hard drive crash or the corruption of a file.

Pendragon Forms is designed for central management on the PC. In order to synchronize a form on your handheld, each Form ID# on the handheld must exactly match a form in the database. If you lose the database that is stored on the PC, you will not be able to synchronize your handheld.

Backing Up the Pendragon Forms Database

The Pendragon Forms database contains all of your form designs. If you are using the Pendragon Forms database for storing your records (as opposed to linking to an external database), then the Pendragon Forms database also contains all of your data.

On a daily basis, you should back up the Forms3.mdb file (or Forms32k.mdb if you are using Microsoft Access 2000), which as noted earlier is typically stored in the C:\Program Files\Forms3 folder. The Forms3.mdb (or Forms32k.mdb) file is the Access database that contains the Pendragon Forms Manager source code, plus your form designs and data.

On a monthly basis, you should back up the entire C:\Program Files\Forms3 folder and its subdirectories.

To protect against a hard drive failure, you should put a copy of your backup on an external drive such as a Zip disk or tape. The size of the Pendragon Forms database starts out at about 3MB, and grows as you create new form designs and collect data. If you are on a local area network, consult your network administrator to see if automatic daily backups can be made.

Backing Up Form Designs

Pendragon Forms requires that for a handheld to synchronize, each form design on the handheld must match the Form ID# of a form in the Forms database.

To see what the Form ID# of a form is, click the name of the form in the Forms Manager and then click the Properties button. The Form ID# is visible in the ID field of the Form Properties window.



To check what a Form ID# is on the handheld, tap the name of a form in the Forms List, then tap the handheld Menu button. Tap the Help menu, then tap Form Info. (Alternatively, tap the form name and use the Graffiti shortcut /I.)

As a precaution, whenever you distribute a form to the handheld, Pendragon Forms makes a backup of the form design in a .pff file. The .pff files are stored in the C:\Program Files\Forms3\PilotF folder.

The relationship between the Form ID# and the name of a .pff file is that the .pff filename is the hexadecimal equivalent of the Form ID#. In the Form Properties window, you can see the name of the .pff file in the field labeled ASCII File. (The ASCII file references an .out extension – this is no longer used, although the filename is still used in conjunction with a .pff extension.)

If you accidentally delete a form design from the database but you still have the form on the handheld and you need to continue to synchronize, you can import the corresponding .pff file to recover the form design. (See cross-reference that follows.)

You can also choose to export form designs and Lookup Lists yourself, instead of relying on the automatic generation of a .pff file. The benefit of exporting a form design and its associated Lookup Lists is that you can select your own name for the exported file, rather than having to use the hexadecimal name of the .pff file.



Refer to Chapter 10, "Managing Form Designs," for information on how to recover a form design and how to export a form design.

Backing Up Data within a Form

In addition to backing up the entire Pendragon Forms database, you may want to make a backup of the data records within an individual form.

To make a backup of the data within a form, you can export the data to ASCII and save the ASCII file to a Zip disk or floppy disk, using the methods described earlier in this chapter.

Summary

This chapter showed you ways to manage your data in the Pendragon Forms database. The bi-directional synchronization property of Pendragon Forms was used to create records on the PC to send to the handheld, such as in a work order scenario. You also saw ways that you could take the data in a form and create reports using tools such as Microsoft Access, Microsoft Word, and Microsoft Excel. This chapter also identified the files that you should back up on a regular basis in order to protect your work.

Chapter 10

Managing Form Designs

IN THIS CHAPTER

- ◆ Learning where form designs are stored
- ◆ Importing data from an old form design into a new form design
- ◆ Importing forms from a different Pendragon Forms database
- ◆ Importing and Exporting form designs and Lookup Lists

FROM THE POINT OF view of your development time, form designs are the most important part of Pendragon Forms. Once you have created a form design that meets your needs, you can use the same form design for months or years.

Working with and protecting your form designs is therefore an integral part of using Pendragon Forms.

Where Are Form Designs Stored?

Form designs are stored internally in the Pendragon Forms database, Forms3.mdb. (If you are using Microsoft Access 2000, the database is Forms32k.mdb.)

In addition, as a safety precaution, whenever you distribute a form to the handheld, a copy of the form design is created in a separate file, as follows:

- ◆ Form design files have a .pff file extension.
- ◆ PFF files are stored in the C:\Program Files\FORMS3\PilotF directory.

The name of a .pff file is related to the database table name for the form. In fact, the .pff filename is the hexadecimal equivalent of the database table name.

To find out the .pff filename for a given form, click the name of a form in the Pendragon Forms Manager, and then click the Properties button. You will see an ASCII File field with an ASCII filename and an .out extension. (The .out file was used in previous versions of Pendragon Forms for storing data from the handheld.) The same filename, but with a .pff extension, is the name of a form design.

Another way to check the .pff file name is to tap the name of a form on the handheld, and then use the Graffiti shortcut /I. (Or tap the handheld Menu button, and then select the Help menu and choose Form Info.)

Synchronizing to a Specific Form Design

Every frozen form has a unique ID number, which is the numeric part of the name of the database table for the form. A form design on a handheld will synchronize with the PC only if a database table with the identical form ID exists in the database.

If you need to synchronize one handheld on two PCs, therefore, it is necessary to have the same form with the same ID number on both PCs as well as on the handheld.

To do this, design the form on one PC, export the form design to a .PFF file, and import the .PFF file into Pendragon Forms on the second PC.



Refer to Chapter 15, "Planning a Multi-User Installation," for issues involved with synchronizing one handheld on two PCs.

If you delete a form design from the database, but the form is still on the handheld, you will not be able to synchronize the form. The best way to recover from this situation is to import the .pff file for the form from the C:\Program Files\Forms3\PilotF folder. This restores the form design and the form ID number to the database.

To determine which .pff file to import, tap the name of the form in the Forms List on the handheld or use the Graffiti shortcut /I. (Alternatively, tap the handheld Menu button, select the Help menu, and then tap the menu option Form Info.) The ASCII filename that is displayed on the handheld screen corresponds to the .pff filename. Once you know the .pff filename, import the .pff file into the Forms Manager on the PC. Refer to the topic "Importing Form Designs and Lookup Lists" in this chapter for specific instructions on importing a .pff file.

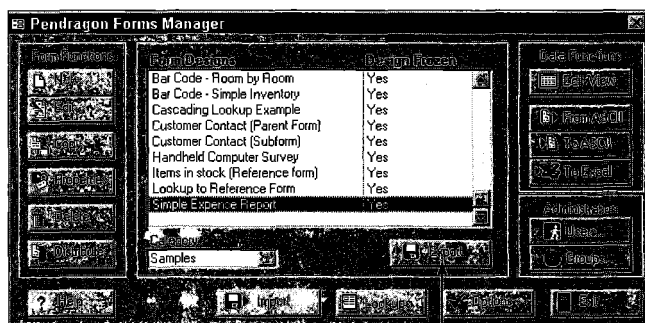
Exporting Form Designs and Lookup Lists

You can choose to export your form designs for backup purposes, or if you need to give a form to another Pendragon Forms user.

Lookup Lists are stored separately from form designs. If your form contains Lookup Lists, you will need to export the Lookup Lists separately from the form designs.

Exporting a Form Design

In the Pendragon Forms Manager, click the name of a form and then click the Export button, as illustrated in Figure 10-1.



The Export button is used to export form designs

Figure 10-1: The Pendragon Forms Manager window

An Export Data window appears. Select the directory where you want to store the form design, and type a name for the file that will store the form design. Click the Save button to save the file.

Figure 10-2 shows a form design being stored in the My Documents folder.

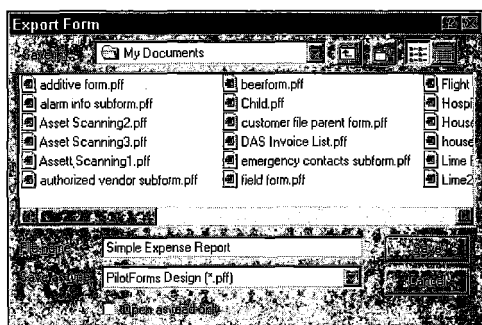


Figure 10-2: Exporting a form design

Form designs are stored in an ASCII format with a .pff file extension. Once you have saved a form design, you can back up the .pff file to a floppy disk or other external medium as needed.

Exporting a Lookup List

If your form design contains Lookup Lists, you will need to export the Lookup Lists separately from the form designs, because Lookup Lists are not stored with form designs.

In the Pendragon Forms Manager, click the Lookups button. The Lookup Editor window will appear, as shown in Figure 10-3. Click the < or > buttons until the Lookup List that you want to export is displayed.

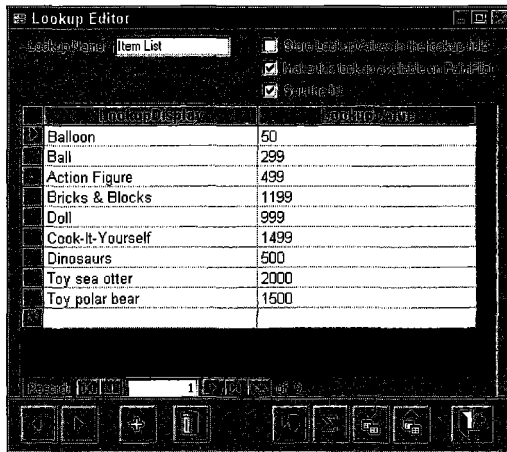


Figure 10-3: The Lookup Editor window

Click the Export Lookups button. An Export Lookup To window appears, as shown in Figure 10-4. Select the directory where the Lookup List is to be stored, and type a filename for the exported Lookup List. Click the Save button.

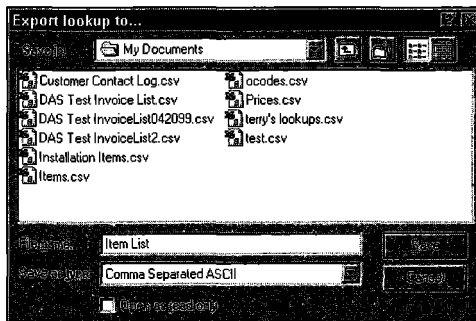


Figure 10-4: Exporting a Lookup List

The Lookup List will be stored as an ASCII CSV (comma-separated variable) file. Repeat the export process for each Lookup List on your form.

Using Import Features

Pendragon Forms offers several import features to enable you to import form designs and data. In the Pendragon Forms Manager, click the Import button to access the Select Import Source window, as shown in Figure 10-5.

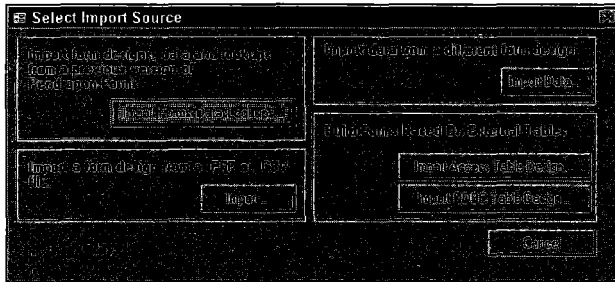


Figure 10-5: The Select Import Source window

Import features are:

- ◆ Importing form designs, data, and Lookup Lists from a previous version of Pendragon Forms
- ◆ Importing a form design from a .pff file format
- ◆ Importing data from one form to another
- ◆ Importing a form design from an external Microsoft Access database table or ODBC database table

The first three features are described in this chapter, and the fourth is explained in Chapter 13, “Linking to an External Access Database,” and Chapter 14, “Linking to an ODBC Database.”

Importing from a Previous Pendragon Forms Database

If you have been using an earlier version of Pendragon Forms, you can import your form designs, data, and Lookup Lists into the current Forms database.

In the Pendragon Forms Manager, click the Import button and then click the Import Forms/Data/Lookups button.

An Import Existing Forms Database window, shown in Figure 10-6, prompts you to select the Forms database to open.

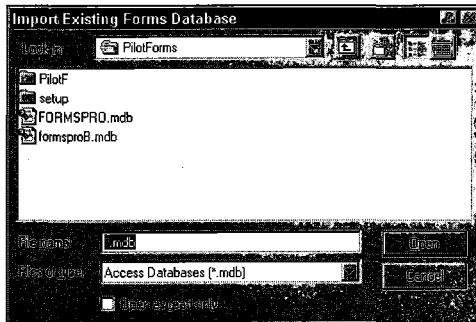


Figure 10-6: The Import Existing Forms Database window

Pendragon Forms version 3 uses a database called Forms3.mdb (or Forms32k.mdb if you are using Microsoft Access 2000).

If you have been using Pendragon Forms version 2, valid database names to select are formsproB.mdb, formsproA.mdb, or formspro.mdb.

If you have been using Pendragon Forms version 1.2, the database name is PilotD7A.mdb.

Select a database and click Open.

As shown in Figure 10-7, a window will display a list of form designs in the selected database. If you have selected the wrong database from which to import, click the Select Database button to reselect the database.

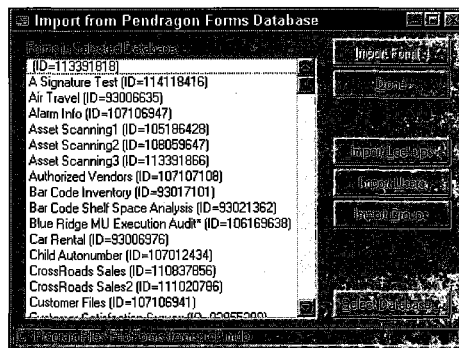


Figure 10-7: The list of forms that can be imported

- ◆ To import a form design and the data in that form, click the name of a form and then click the Import Form button. Repeat for all form designs that you want to import.
- ◆ To import all of your Lookup Lists, click the Import Lookups button. You need to do this only once.

- ◆ If you are importing from Forms version 3, you can click the Import Users button and the Import Groups button to import your User List and User Groups.

Importing Data from a Different Form Design

When you copy a form to make changes to the form design, the data from the original form design is not automatically copied into the new form.

If you have not substantially changed the new form – for example, you have not changed field types of existing fields – you can copy the data from the original form into the new form, as follows:

1. Make the necessary modifications to the new form.

Ideally you should not change column names on the Advanced Field Property screen. If you keep the column names of the new form the same as on the old form, Pendragon Forms can easily determine which field from the old form should map to each field on the new form.

2. Click the name of the copied form, click the Import button, and then click the Import Data button.

Deploying Pendragon Forms to Another PC

If you have been designing forms on one PC and then need to deploy the program to another PC, you can export your form designs and Lookup Lists from the first PC, and import them into Pendragon Forms on the second PC. (Note that a license is required for each Palm device on which Pendragon Forms is installed.)

If you have a lot of form designs and Lookup Lists, the easiest way to move all of your form designs and Lookup Lists to the second PC is to make a copy of the Forms3.mdb database (or Forms32k.mdb if you have Microsoft Access 2000). Give the copy a different name. Place the copy in a shared directory accessible to the second PC, or, in the case of a stand-alone PC, place the copy in a temporary folder in Windows Explorer.

In the Pendragon Forms Manager on the second PC, use the procedure for importing forms from a previous Pendragon Forms database to import your form designs and Lookup Lists from the copied database. The benefit to this approach is that you can import all of your Lookup Lists in one step, without having to worry about individual files.

- An Import Existing Form Data window appears, as shown in Figure 10-8. In the Import from Form field, select the form whose data you want to import.

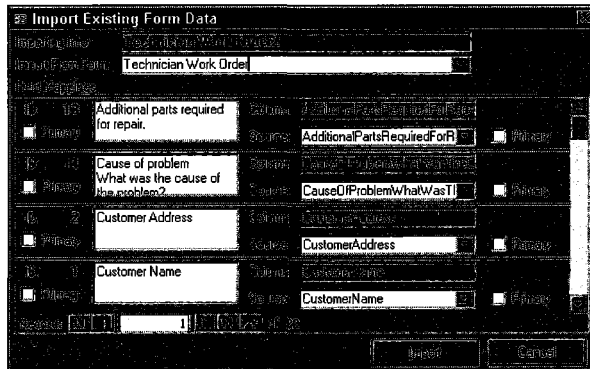


Figure 10-8: The Import Existing Form Data window enables you to copy data from an old form design into a new form design

- Once the original form has been selected, a list of fields will be displayed.

If the column names of the new form are the same as those of the old form, click the Import button to map data from the original form to the new form.

If the database column names were changed in the new form, then select the old column name in the Source field for each field on the form.

If you accidentally map a Text field from the old form to a Yes/No field or a Popup List in the new form, data for these fields will not be sent to the handheld.

Importing Form Designs and Lookup Lists

If you have exported form designs and Lookup Lists to ASCII for backup purposes or to move the forms from one PC to another, you can import these files into the Pendragon Forms Manager when needed.

Each form has a unique form ID number that is part of the name of the database table for the form. When you import a form design, if the form ID number for that

form does not currently exist, Pendragon Forms will keep the same form ID number in the newly created database table for that form. This makes it possible to synchronize the same form design on a different PC from the PC on which the form was designed.

If the form ID number associated with an imported form already exists in the database, Pendragon Forms will assign a different form ID number. This means that from the point of view of synchronizing, the imported form is considered different from the original form.

Importing a Form Design from a .PFF File

In the Pendragon Forms Manager, click the Import button to display the Select Import Source button. Then click the Import button.

An Open window appears, as shown in Figure 10-9. Pendragon Forms Design files (*.pff) will be displayed by default.

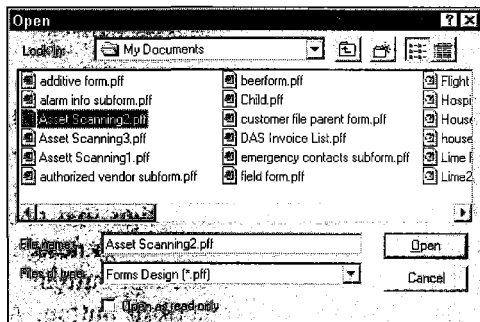


Figure 10-9: Importing a Form Design

Select the directory where the .pff file is stored, and select the name of the file to be imported. Click the Open button to import the form design into the Pendragon Forms Manager.

If the form that you are importing contains Lookup Lists, you will need to import the Lookup Lists separately.

Importing Lookup Lists

To import a Lookup List, click the Lookups button in the Pendragon Forms Manager. The Lookup Editor window, shown in Figure 10-10, is displayed.

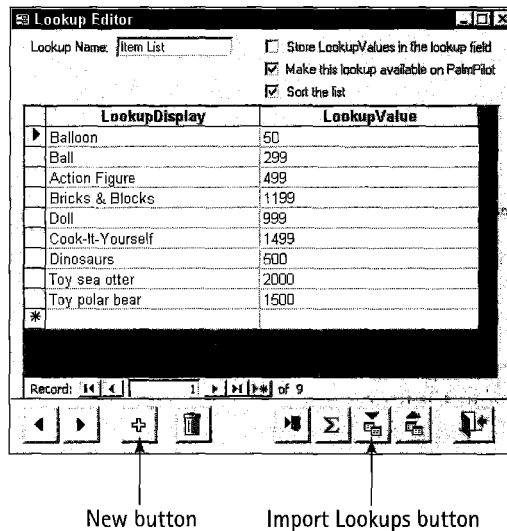


Figure 10-10: The Lookup Editor window

Click the New button (+) to create a new Lookup List. In the Lookup Name field, you will need to type the same name for the Lookup List as was selected in the Lookup List field on the form.

If you do not know what name to use, you can edit the form design first and check the name of the Lookup List referenced on the form. To edit a form design, click the name of the form in the Forms Manager and then click the Edit button.

After typing a name for the Lookup List, click the Import Lookup button. A Lookup to Import... window, shown in Figure 10-11, will be displayed. Select the directory where the Lookup List ASCII CSV file is stored, and select the filename. Click Open to import the Lookup List.

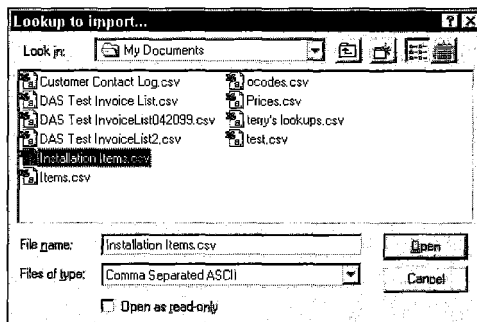


Figure 10-11: Importing a Lookup List



You can also use the Import Lookup feature to create a Lookup List from data in an ASCII format. The ASCII file should be in comma-delimited (also called comma-separated variable) CSV format. Because a Lookup List typically has one column of data, the CSV file should have only one column of data as well (or two columns if you are using Lookup Values).

Summary

In this chapter you saw how to import form designs from one Pendragon Forms database to another, and how to move form designs from one PC to another. You also learned that form designs are stored separately from Lookup Lists, and so Lookup Lists have to be imported and exported in addition to form designs. Most important, you learned that by exporting form designs, you can back up your work.

Part IV

Creating Specialized Forms

IN THIS PART

CHAPTER 11
Using Scripts

CHAPTER 12
Using Bar Codes

Chapter 11

Using Scripts

IN THIS CHAPTER

- ◆ Writing scripts
- ◆ Performing calculations
- ◆ Creating forms that branch
- ◆ Programming button fields

What Is a Script?

A *script* is a sequence of commands that perform actions based on input from the handheld user. Scripts can be used to perform calculations on fields that the handheld user enters and can be used to create branching, so that a selection in one field determines which fields are displayed next.

Scripts are the means by which you can program a Button field, control bar code scanner input on an SPT 1500, and send data to a Web site via a Palm VII.

Figure 11-1 shows a simple form for calculating mileage expenses. A script in Field 5 calculates the total miles traveled. Another script in Field 7 calculates the reimbursement amount.

Sales call	
Purpose of Trip	Sales call
Date	11/9/99
Starting Mileage	
Ending Mileage	
Total Miles	0
Per mile Reimburse	\$0.00
Amount of Reimbu	\$0.00

End ⏪ ⏩ ⏴ ⏵

Figure 11-1: A Mileage Log form

Scripting is an Advanced Field Property. To add a script in Field 5, for example, display Field 5 in the Form Designer window, and then click the Advanced Field

Properties button, as shown in Figure 11-2. In the Advanced Field Properties window, click the Script tab.

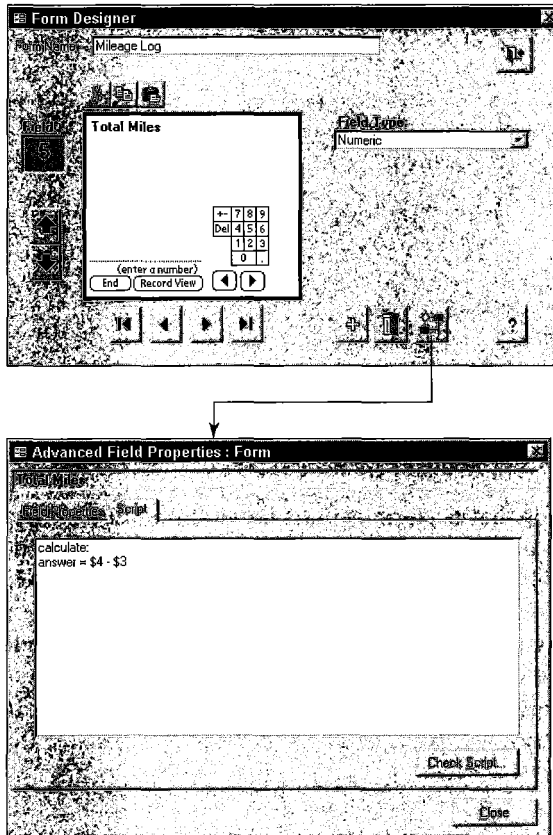


Figure 11-2: Click the Advanced Field Properties button to access the script for a given field.

Every script consists of at least one *event procedure*. The event procedure contains an event procedure name and one or more statements. The event procedure name determines when the script will be run. The *statements* are the actions that are performed when the script runs.

As shown in Figure 11-3, the Total Miles amount in Field 5 is calculated by subtracting the Starting Mileage (Field 3) from the Ending Mileage (Field 4).

To perform the script in Field 5, enter the following in the Script field of the Advanced Field Properties window for Field 5:

```
calculate:
answer = $4 - $3
```

Sales call	
Purpose of Trip	Sales call
Date	11/9/99
Starting Mileage	48252
Ending Mileage	48409
Total Miles	157
Per mile Reimburse	\$0.00
Amount of Reimbu	\$0.00

End ⏪ ⏩ ⏴ ⏵

Figure 11-3: Entering a Starting Mileage and an Ending Mileage runs a script to calculate Total Miles.

A calculate event procedure is typically used to perform calculations. A calculate event runs whenever the user exits a changed field. This means that whenever the handheld user enters or modifies a value in a field, the script updates the calculated result.

A dollar sign in front of a number is a field reference that refers to the value in the specified field. For example, \$4 means the value in Field 4. The expression \$4 - \$3 means the value in Field 4 minus the value in Field 3.

The word “answer” is a variable that represents the value of the current field. The statement `answer = $4 - $3` subtracts the value in Field 3 from the value in Field 4 and then places the calculated result in the current field. In the Mileage Log example, because the script is written in Field 5, “answer” refers to Field 5.

To calculate the reimbursement amount, as shown in Figure 11-4, a script in Field 7 multiplies the Total Miles (Field 5) by the reimbursement amount per mile (Field 6).

Sales call	
Purpose of Trip	Sales call
Date	11/9/99
Starting Mileage	48252
Ending Mileage	48409
Total Miles	157
Per mile Reimburse	\$0.25
Amount of Reimbu	\$39.25

End ⏪ ⏩ ⏴ ⏵

Figure 11-4: Another script is used to calculate the Reimbursement Amount for the miles traveled.

The script in Field 7 is:

```
calculate:
answer = $5 * $6
```

Once you have written a script, click the Check Script button in the Advanced Field Properties window. The Script Checker verifies that the syntax of your script is correct, as shown in Figure 11-5. Running the Script Checker reduces the chance of errors on the handheld when the script is run.

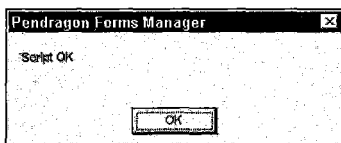


Figure 11-5: Click the Check Script button to check the syntax of your script.



You can modify scripts even after a form has been frozen. Make your changes to a script and then save the form design and redistribute the form to the handheld.



When designing a form, it is best to create all the fields on the form before you write the scripts in the various fields. If you reference a field in a script as Field 4, and then you later insert a field so that Field 4 becomes Field 5, the script will not automatically update to change the reference. You will need to manually check all field references in all scripts if you insert fields on your form.



For information on errors that the Script Checker detects, see Appendix A, "Troubleshooting Tips."

How Scripting Works and Limits of Scripts

Scripts are transferred to the handheld in ASCII format. When a form first needs to use a script, the script is compiled and then run. The compiled version of the script then replaces the ASCII format on the handheld.

Scripts are compiled for a given form when the first new record is created on the handheld. In the case of records being populated from the PC, scripts are compiled when the first record is reviewed on the handheld. If you redistribute a form to the handheld, the scripts associated with that form are not initially compiled but are later compiled when needed.

The text in each script cannot exceed 2,000 characters. A carriage return (pressing Enter) counts as two characters.

The computer processor inside the handheld is not very fast – perhaps less than 1 percent of the speed of a typical desktop PC. This means that if you write long scripts, there will be a noticeable delay in moving from one field to another as the processor runs the script. To create forms that are usable on the handheld, therefore, it is important to limit the use of scripts and to minimize the length of scripts.

Using Event Procedure Names

Each event procedure starts with an event procedure name that determines when the script will run. Every script must contain at least one event procedure name.

There are eight different event procedure names. As shown in Table 11-1, some events work only in Field View (viewing one field at a time), and others work both in Field View and in Record View (viewing 11 fields at a time).



In a script, a colon must follow an event procedure name.

TABLE 11-1 SCRIPTING EVENT PROCEDURE NAMES

Event Procedure Name	Used For	Works in Field View	Works in Record View
calculate	Performing calculations that automatically update when a field changes	✓	✓

Continued

TABLE 11-1 SCRIPTING EVENT PROCEDURE NAMES (Continued)

Event Procedure Name	Used For	Works in Field View	Works in Record View
click	Programming Button fields	✓	✓
enter	Branching scripts and bar code scripts	✓	
exit	Branching scripts and bar code scripts	✓	
initialize	Running a script once when a new record is created	✓	✓
open	Running a script every time a record is entered	✓	✓
select	Branching scripts	✓	
validate	Running a script just before leaving a record	✓	✓

Calculate Event

The `calculate` event occurs when you leave any modified field. A `calculate` event is used to place the result of a calculation into a field. Typically, the calculation depends on fields that the handheld user enters, and if the handheld user updates a field, the calculated result is also automatically updated.

All `calculate` events run in sequence. This means, for example, that if Field 5 and Field 6 both contain a `calculate` script, the script in Field 5 will run before the script in Field 6.

Because `calculate` events run whenever any field on a form is updated, there is a performance penalty if the form has a lot of fields. For example, if a form contains 100 fields for data entry, and there is one `calculate` script, the script will run 100 times – once for each field as it is filled in. If you are using Field View for data entry, an `enter` event is faster for calculations, because the calculation is performed only once per record.

Click Event

The `click` event is only used with Button fields. A `click` event runs a script when the handheld user taps the button in the Button field.

Enter Event

The `enter` event occurs when a field is about to be drawn. This means that the script is run when the user displays the field on the handheld.

An `enter` event runs before the handheld user inputs any data in a field. An `enter` script can be used to place a calculation in a field, or to display or hide subsequent fields on the form, or to specify which bar codes a user can scan into a field.

Exit Event

The `exit` event occurs when the handheld user exits a field – that is, when the user taps the Next button (right arrow button) to move to the next field on a form. An `exit` script runs only when the you are moving “forward” – for example, from Field 3 to Field 4, and not “backward,” from Field 4 to Field 3.

The `exit` event is ideal for use in branching scripts. As the user exits a field, the `exit` script can look at the user’s response in the current field and determine which field to display next.

Initialize Event

The `initialize` event runs only once, when a new record is created. An `initialize` script is typically used to fill in values in certain fields, so that when the handheld user creates a new record, some fields are prefilled.

The following `initialize` script is commonly used to place the current date and time in a Date & Time field:

```
initialize:  
answer = now
```

Open Event

The `open` event runs once whenever a record is entered. Whereas an `initialize` event runs only when a new record is created, an `open` event will run both when a record is new and whenever the record is reviewed.

The `open` event can be used to set up whether fields on a form are hidden or displayed when the user enters a record. An `open` event can also be used to set up bar code parameters, such as whether the bar code scanner is enabled or not.

Select Event

The `select` event occurs when the handheld user selects an option or accepts a dialog associated with a field. The `select` event is typically used with branching scripts, so that the user’s selection in a field determines the next field to be displayed.

One limitation of a `select` event is that the script does not run if the handheld user does not make a selection in the field. If there is a possibility that the handheld user will leave a field blank, an `exit` script can be used to handle the case when the field is blank.

Validate Event

A `validate` event occurs when the handheld user leaves a record. A `validate` event typically checks that the user has entered a correct or unique value in a field.

Using Variables

Variables are used in statements to represent values in calculations, and to refer to the values of fields.

Pendragon Forms scripts support the following variables:

- ◆ `answer`: Represents the value of the current field – that is, the field in which the script is written. For example, to set the value of the current field to the number 10, the following script can be used:

```
enter:  
answer = 10
```

To add Field 5 and Field 8, and place the result in the current field, the following script can be used:

```
calculate:  
answer = $5 + $8
```

- ◆ `result`: Used as a temporary variable for storing intermediate calculations. For example, a script to calculate the average of two numbers is:

```
calculate:  
result = $6 + $7  
answer = result / 2
```

- ◆ `$number`: `$` is a field reference that refers to the value in the specified field. The reference `$3` means the value in Field 3. A script to multiply the value in Field 5 by 100 and place the result in the current field is shown here.

```
calculate:  
answer = $5 * 100
```

- ◆ `now`: This is a function that is used to store the current date and time. A script to record the creation date and time of a new record is as follows:

```
initialize:
answer = now
```

A script to record the last modification date and time of a record is shown here.

```
calculate:
answer = now
```

- ◆ **null**: This is a constant that is equivalent to an empty string. The following script does nothing if Field 8 is blank but divides Field 4 by Field 8 otherwise.

```
calculate:
if $8 = null then
return
endif
answer = $4 / $8
```

Using Operators

The Pendragon Forms scripting language contains unary and binary operators that make it possible to perform calculations.

Unary Operators

Unary operators act on a single value. There are three unary operators.

- ◆ **integer value**: Converts a number to its integer (whole number) part. For instance, 15.7 is converted to 15. Negative numbers are converted downward; for example, -22.5 is converted to -23. The following script places the integer part of Field 18 into the current field:

```
calculate:
answer = integer $18
```

- ◆ **- (Minus sign)**: Changes the sign of a number. A positive number converts to a negative number and vice versa. For example, this script converts the sign of the value in Field 4 and places the result in the current field:

```
calculate:
answer = -$4
```

- ◆ **length value**: Returns the number of characters in a value. For example, this script returns the number of characters in Field 12:

```
calculate:
answer = length $12
```

Binary Operators

Binary operators combine two values. Pendragon Forms supports seven binary operators.

- ◆ **+** (Addition): Adds two values. For example, to add the values of Field 20 and Field 25, this script can be used:

```
calculate:  
answer = $20 + $25
```

- ◆ **-** (Subtraction): Subtracts one value from another. This script subtracts Field 6 from Field 7:

```
calculate:  
answer = $7 - $6
```

- ◆ ***** (Multiplication): Multiplies two values. This script multiplies the values of Field 15 and Field 3:

```
calculate:  
answer = $15 * $3
```

- ◆ **/** (Division): Divides one value by another. This script divides the value in Field 9 by the value in Field 10:

```
calculate:  
answer = $9 / $10
```

- ◆ **#** (Contains): Looks for a character string. For example, this script branches to Field 6 if the current field contains the word Red and branches to Field 8 if the current field contains the phrase Blue Green. Double quotation marks are required if a string contains spaces.

```
exit:  
if answer # Red then  
goto 6  
endif
```

```
if answer # "Blue Green" then  
goto 8  
endif
```

- ◆ **%** (Modulo division): Returns the remainder of a division. This script returns the remainder after dividing Field 5 by the number 3:

```
calculate:  
answer = $5 % 3
```

If Field 5 contained the number 11, for example, the value placed into the current field would be the number 2.

- ◆ **& (Concatenate):** Combines two strings. This script concatenates the value in Field 2 with the phrase “miles per hour”:

```
calculate:  
answer = $2 & " miles per hour"
```

If Field 2 contained the number 65, the string in the current field would be 65 miles per hour.

Using Statements

A statement is a command that is performed when a script is run. Three types of statements can be used in scripts: assignment statements, conditional statements, and action statements.

An event procedure can contain one or more statements. For example, an `exit` event can cause a calculation to be performed, several fields to be hidden or shown, and a branching effect to occur. The different statements are all triggered by the `exit` event.

Using Assignment Statements to Place a Value in a Field

Assignment statements place a value into a field or into a temporary variable.

Pendragon Forms supports only simple expressions involving not more than two values and one operator per assignment statement.

The four assignment statements are:

- ◆ `answer = <expression>`: Places the result of a calculation into the current field. In this example, Field 7 is divided by the number 10, and the calculated result is placed into the current field.

```
calculate:  
answer = $7 / 10
```

In the following example, Field 3 is multiplied by Field 5, and the result is placed into the current field:

```
calculate:  
answer = $3 * $5
```

- ◆ `result = <expression>`: `result` is a temporary variable. Because not more than two values can be calculated at a time, `result` is used as extra “storage space” for intermediate calculations. For example, to add the

values in Fields 8, 9, 10, and 11, and to place the calculated result in the current field, the following script can be used:

```
calculate:
result = $8 + $9
result = result + $10
answer = result + $11
```

For a calculation such as (Field 3 + Field 4) / 12, the following script can be used:

```
calculate:
result = $3 + $4
answer = result / 12
```

- ◆ `$number = <expression>`: This is used to assign a value to a field from a script in another field. In the following example, Field 6 records the current date and time when the handheld user exits the current field.

```
exit:
$6 = now
```

- ◆ `assign`: Assigns a value to the current field. Similar to the `answer` statement, but `answer` is preferred because an `answer` statement is less ambiguous and easier to read. The following script sets a default value of 100 in a Numeric field:

```
calculate:
if answer = null then
assign 100
endif
```

Testing the User's Response with a Conditional Statement

There is only one *conditional statement*, the `if . . . then . . . endif` statement. The conditional statement is used to make a decision depending on the response that the handheld user enters in a field. Tests for a condition enable you to take action if the condition is true.

The format of the conditional statement is:

```
if condition then statements endif
```

The following conditional operators can be used in an `if` statement.

=	Equals
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
<>	Not equal to
#	Contains (used with strings)

The conditional operators are all binary operators that take two arguments.

In the following example, a conditional statement is used to test if the current field is blank. If the field is blank, a value of No is assigned to the field.

```
exit:
if answer = null then
  answer = No
endif
```

When writing an `if` statement, include both the actions that are performed if the condition is true and the actions to be performed if the condition is false.

In the following example, a test is performed to see whether Field 5 is greater than or equal to 100. If this condition is true, Fields 18, 19, and 20 are made visible, and the script branches to Field 18. If Field 5 is less than 100, the script branches to Field 21.

```
exit:
if $5 >= 100 then
  show from 18 to 20
  goto 18
endif

goto 21
```

The `#` (Contains) conditional operator is used to test whether a character string is present. If the character string is one word, no quotation marks are needed. If the character string includes spaces, double quotation marks are required around the phrase, as shown here.

```
select:
if answer # Basic then
  goto 6
endif

if answer # "Intermediate Level" then
  goto 10
endif

goto 15
```

Using Action Statements in a Script

Action statements enable a script to perform an action. Action statements are often used with a conditional statement, so that if a particular condition is true, the actions are performed.

Action statements can make a field visible or hidden, required or optional, read-only or able to be updated. Action statements are also used in branching, to jump the handheld user to a particular field.

Specialized action statements include performing a lookup to the Palm Address Book, printing, performing a lookup to a Lookup List, and attaching a record via an e-mail message.

The Action statements are described here.

- ◆ **abortform:** Aborts the form, exiting the script and deleting the current record from the handheld device. Can be used in `enter`, `select`, and `exit` events (that is, in Field View only). In the following example, the record is deleted and the form is aborted if Field 2 is blank. (Note that the `endform` statement in this example is the action that is performed if Field 2 is not blank.)

```
exit:
if $2 = null then
  abortform
endif
endform
```

- ◆ **beep:** Sounds a standard warning beep. In the following example, if the handheld user exits the current field without entering any data, the Palm organizer will beep.

```
exit:
if answer = null then
  beep
endif
```

- ◆ **count *field-number***: Checks the current record against all the other records in the database, to see how many records have the same value in the specified field. The number of matching records is placed in the result variable. For performance reasons, the count statement should not be used in a calculate event. The following example checks to see if Field 6 is unique and displays a message if not:

```
validate:
count 6
if result > 0 then
  msgbox "This is a duplicate barcode"
endif
```

- ◆ **endform**: Equivalent to pressing the End button. Exits the script, saves the record, and then exits the form. This statement is available only in enter, select, exit, and click events. In the following example, if the answer in the current field contains the word "Done," the record is saved and the form is exited.

```
exit:
if answer # Done then
  endform
endif
```

- ◆ **filtercount *field-number***: Used to check uniqueness in a subform. In a subform, records are filtered to display only those that match the parent record. To check that a subform record is unique, a filtercount script can be added to the subform. The filtercount statement checks how many of the currently filtered records match on the specified field, and the number of matching records is placed in the result variable. For performance reasons, filtercount should not be used in a calculate event. In the following example, a check is performed to see if Field 14 is unique on a subform.

```
validate:
filtercount 14
if result > 0 then
  msgbox "Another record has this value"
endif
```

- ◆ **font {answer | question} {bold | normal | large}**: Works in Field View only to set the font for the field name (question) or the response. Available only for use in enter, select, and exit events. In the following example, the font for a field name is set to bold, and the font for the response is set to large.

```
enter:
font question bold
font answer large
```

Note that if you are working in Field View only, if you set the font for a field name or response, then the font setting is retained for subsequent fields also. If you switch from Field View to Record View and back, however, the font setting is lost on subsequent fields.

- ◆ `formsum "name-of-form" field-number`: Adds up values across all records for the specified form and the specified field and places the calculated result in the `result` variable. For performance reasons, this should primarily be used in `click` events. In the following example, clicking a button adds up all the values in Field 20 of the form "Daily Sales." The calculated result is placed in Field 8 of the current form.

```
click:  
formsum "Daily Sales" 20  
$8 = result
```

- ◆ `getaddress`: Prompts the handheld user to select an address from the built-in Palm Address Book application. Contact information is copied from the Address Book into fields that are named to receive the information. The `getaddress` statement is typically used with a `click` event in a Button field. It is unavailable in a `calculate` event. In order to receive Address Book data, the form must contain one or more of the following fields, named identically as follows (including uppercase letters and a colon):

- First name:
- Last name:
- Title:
- Company:
- Address:
- City:
- State:
- Zip:
- Country:
- Tel:
- Other:
- Email:
- Custom:

Tel refers to the Work phone number, and Custom refers to the Custom 4 field.

In the following example, a `click` event is used in a Button field to enable the handheld user to tap a button to look up an Address Book contact:

```
click:
getaddress
```

- ◆ `goto field-number`: Moves to the specified field on the form and then exits the current script. Can be used only in Field View in `select`, `exit`, and `validate` events. In the following example, the script checks what response was selected when the handheld user exits the current field. If the user selects Y, the form branches to Field 10, and if the user selects N, the form branches to Field 12. If the field is left blank, the form branches to Field 10.

```
exit:
if answer = Y then
  goto 10
endif
```

```
if answer = N then
  goto 12
endif
```

```
goto 10
```

If you use a `goto` statement, you should always code for the instance in which none of the conditions is met. In the preceding example, for example, if the answer is neither Y nor N, the script still forces a branch to Field 10.

- ◆ `hide field-number; hide from field-number to field-number`: Used to hide a single field or multiple fields. Works best in Field View, because in Record View hiding a field does not automatically update the screen to hide the field. The following example hides Field 8 if the answer in the current field is "Below Normal" and hides Fields 9–12 if the answer is "Above Normal."

```
exit:
if answer # "Below Normal" then
  hide 8
endif
```

```
if answer # "Above Normal" then
  hide from 9 to 12
endif
```

Note that in order to reduce the number of scripts on a form, instead of using scripts to hide fields you can set the Advanced Field property of Hidden. Scripts can then just be used to display fields.

- ◆ **invalidate "message":** Works only with the validate event. Flags a record as having an invalid value, and the handheld user cannot exit the record until the invalid value is corrected. In the following example, when the user attempts to exit the record, a message is displayed if the answer in this field is greater than 100. The user has to correct the field in order to exit the record.

```
validate:
if answer > 100 then
  invalidate "Score cannot exceed 100 points"
endif
```

Note that one way to avoid the use of scripts with a Numeric field is to use the Advanced Field Property of Max & Min. This gives the user feedback as soon as the field is exited, instead of waiting until the user attempts to exit the record. (A validate event runs only when the user exits a record, so if a form contains a validate event in Field 3, and 60 fields are on the form, the user will receive the error message only after filling in 57 more fields, instead of immediately after filling in Field 3.)

- ◆ **left value length:** Extracts the left-most characters up to the specified value and places the extracted characters in the result variable. The following example puts the string CJ528 in the current field.

```
enter:
left "CJ5287Z" 5
answer = result
```

The following example puts the first six characters of Field 2 into the current field.

```
enter:
left $2 6
answer = result
```

- ◆ **lookup value within lookup-list-name:** Used to look up a value in a Lookup List via a script. The *value* is the name of the item to look for, and the *lookup-list-name* is the Lookup List to search. The value that is found is placed into the result variable.

A `lookup . . . within script` is typically used if you want to make a selection or scan a bar code into one field and then automatically display the item's name or price in another field. The Lookup List that you are referencing must contain Lookup Values and be set to display the Lookup

Values. This makes it possible to look up an item but display the corresponding Lookup Value in the field.

In the following example, a bar code is scanned into Field 1 of a form. The script in Field 1 looks up the bar code in the Price List and copies the price into Field 2.

```
exit:
lookup $1 within "Price List"
$2 = result
```

- ◆ *lookup value within form-name*: Performs a lookup to another form. The *value* is the name of the item to look for. If a match is found, all the fields from the other form whose names match fields on the current form will be copied into the current form.
- ◆ *mid value start length*: This function copies *length* number of characters from the specified *value*, beginning at position *start*. The extracted characters are placed into the *result* variable. The following example places the string 98765 into the current field:

```
calculate:
mid "ABX98765QTY" 4 5
answer = result
```

- ◆ *msgbox "string"*: Displays a dialog box on the handheld. The following example displays a message if the user enters a value greater than 100 in the current field.

```
exit:
if answer > 100 then
  MsgBox "The temperature is very high"
endif
```

- ◆ *optional field-number; optional from field-number to field-number*: Makes the specified field or fields optional. Used to override the Advanced Field Property of Required. In the following example, if the answer in the current field is Y, then Field 7 is optional. If the answer is N, Fields 8 and 9 are optional.

```
exit:
if answer # Y then
  optional 7
endif

if answer # N then
  optional from 8 to 9
endif
```

- ◆ `print { serial | IR }`: Causes the current record to be printed, either via the serial port or via infrared. Printing from Pendragon Forms via infrared requires additional Bachmann Print Manager software from www.bachmannsoftware.com to be installed on the handheld. (An evaluation version is on the CD-ROM.) The print statement is typically used in a click event so that the handheld user can control when printing occurs. The following example prints a record via infrared when the handheld user taps a button:

```
click:
print IR
```

- ◆ `readonly field-number; readonly from field-number to field-number`: Used to make one or more fields read-only. In the following example, if the handheld user selects the option Completed Task, Field 7 and Fields 10–25 are made read-only.

```
exit:
if answer # "Completed Task" then
readonly 7
readonly from 10 to 25
endif
```

- ◆ `readwrite field-number; readwrite from field-number to field-number`: Used to make a read-only field capable of being updated. In the following example, if Field 8 is less than Field 6, Field 9 can be updated.

```
exit:
if $8 < $6 then
readwrite 9
endif
```

- ◆ `require field-number; require from field-number to field-number`: Makes the specified field or fields required fields. In the following example, if the response “Level 2 Inspection” is selected in the current field, then Fields 18–23 become required.

```
exit:
if answer # "Level 2 Inspection" then
require from 18 to 23
endif
```

- ◆ `return`: Exits from the current script without executing any more instructions. Useful if you want to prevent additional script statements from being performed after an if statement. In the following example, a response of “Basic Level” assigns a score of 500. The return statement then ends the script at this point to prevent further calculation. If the response does not contain “Basic Level,” then the score is set to 1000.


```

exit:
if answer # "Basic Level" then
    $5 = 500
    return
endif

$5 = 1000

```

- ◆ **reverselookup *value* within *lookup-list-name***: This is the opposite of the `lookup . . . within` statement. Instead of looking up an item and finding the corresponding Lookup Value in the specified Lookup List, `reverselookup . . . within` looks up the lookup value and places the corresponding lookup display item in the result variable.
- ◆ **right *value length***: Extracts the right-most characters of the specified length and places them in the result variable. The following example places the characters 347 into the current field.

```

enter:
right "Procedure Code 347" 3
answer = result

```

- ◆ **show *field-number*; show from *field-number* to *field-number***: Makes a hidden field or range of fields visible. Works best in Field View, because Record View does not automatically update the handheld screen to show or hide fields. In the following example, if the handheld user selects Other in the current field, Fields 4 through 6 are made visible.

```

exit:
if answer # Other then
show from 4 to 6
endif

```

- ◆ **transmit multimapil "*Email address*"**: Used in conjunction with Multi-Mail Pro e-mail software from www.actualsoft.com. (An evaluation version of MultiMail is on the CD-ROM.) Creates a text version of the current record and adds it to the e-mail Outbox to be sent to the specified e-mail address when the handheld next sends e-mail. Recommended for use in click events only, to give the handheld user control of when e-mail is sent. In the following example, the current record is e-mailed to `sales@xyz.com`.

```

click:
transmit multimapil "sales@xyz.com"

```

In the next example, the current record is e-mailed to the e-mail address in Field 4:

```

click:
transmit multimapil $4

```

- ◆ `transmit palmnet "address"`: Used in conjunction with the Palm VII. Refer to Chapter 16, "Working with the Palm VII."

Using Action Statements with the SPT 1500

Certain scripting action statements have been designed for use with the Symbol Technologies SPT 1500 bar code scanning solution.



Refer to Chapter 12, "Using Bar Codes," for information on action statements for controlling the bar code scanner of an SPT 1500.

Creating Calculation Scripts

To perform a calculation, add a script to the field in which you want to display the result of the calculation. The field that stores a calculated result is typically a Text field, a Numeric field, or, if you are working with Currency amounts, a Currency field.

The `calculate` event is typically used for calculation scripts, because a `calculate` event runs both in Field View and in Record View and automatically runs whenever any field on the form is changed.

One alternative to using a `calculate` event is to use an `enter` event if you are only using Field View for data entry. On a form with a lot of fields, an `enter` event is more efficient because the calculation runs only once when you enter the field, not every time every field is modified.

Another alternative to using a `calculate` event is to create a Button field on your form and to perform the calculation only when the handheld user taps the button.



If you do not want the handheld user to be able to directly modify a calculated result, set the Advanced Field Property of Read-Only in the field that is storing the result. See Chapter 6, "Advanced Field Properties."

Working with Several Fields in a Calculation

Each statement in a script can support only simple expressions involving a maximum of two values. To add the values in Fields 2 and 3, for example, the following script can be used:

```
calculate:
answer = $2 + $3
```

A calculation is more complex when more than two values have to be calculated. In Figure 11-6, for example, a calculation is required to add the values in five fields.

9/27/99	
Week Beginning	9/27/99
Monday Hours	7.5
Tuesday Hours	8
Wednesday Hours	6.5
Thursday Hours	8
Friday Hours	7.5
Total Hours	37.5

End ⏪ ⏩ ⏴ ⏵

Figure 11-6: The calculation script in the Total Hours field has to add five values together.



If you installed Pendragon Forms from the CD-ROM, open the Pendragon Forms Manager and select the Samples category. For an example of the form shown in Figure 11-6, distribute the form Hours Worked on the CD-ROM to the handheld.

To add the number of hours worked in a week, the `result` variable is used to store intermediate calculations in the script. It is possible to add the current value of the `result` variable to another number and then store a new result in the `result` variable. The `answer` variable is used to store the final calculated result into the current field. The script in Field 7, Total Hours Worked, is as follows:

```
calculate:
result = $2 + $3
result = result + $4
result = result + $5
answer = result + $6
```

Dividing

When you are dividing one number by another, you need to be careful to avoid division by zero errors. Dividing a number by zero causes an error on the handheld because the result cannot be represented on the Palm computer.

If you need to divide two numbers, as shown in Figure 11-7, the way that calculate scripts run causes the potential for a divide by zero error. Because a calculate script runs as soon as a field changes, if you enter a value for Number A on the form, the calculate script in Field 3 runs as soon as you leave Field 1. However, because no value has yet been entered for Field 2, the script runs the division Number A / 0, which causes a division by zero error.

Figure 11-7: A form for dividing two numbers

The error message that appears when a division by zero error occurs is shown in Figure 11-8.

Figure 11-8: An error message is generated on the handheld if a calculate script attempts to divide one number by zero.

To prevent the division by zero error, the calculate script can include a simple test to prevent running the script if Number B (Field 2) is zero. The script in Field 3 is:

```
calculate:
if $2 = 0 then
  return
endif

answer = $1 / $2
```

If Field 2 is zero, the return statement ends the script without performing the calculation. Only if Field 2 contains a number will the division take place in Field 3.

Rounding

When you divide one number by another, the handheld displays as many decimal places as possible. In some instances you may prefer to round your answer to a specific number of decimal places.

Figure 11-9 shows a form that is used to convert degrees Fahrenheit to Celsius. In Field 2, the script to convert Fahrenheit to Celsius is:

```
calculate:
result = $1 - 32
result = result * 5
answer = result / 9
```

Because a division occurs in Field 2, the value that is displayed in this field shows as many decimal places as possible on the handheld.

Fahrenheit #	55
Celsius Equivalent	12.777778
Celsius - 2 dec. plac	12.78
Celsius - whole num	13

Figure 11-9: This form illustrates a division without rounding (Field 2), then rounding to two decimal places (Field 3), and rounding to a whole number (Field 4).

To round the answer in Field 2 to two decimal places, the following script is used in Field 3:

```
calculate:
result = $2 * 100
result = result + 0.5
result = integer result
answer = result / 100
```

Multiplying by 100 preserves two decimal places of precision. Adding 0.5 rounds the number up if the remaining decimal part of the number is greater than 0.5. The

integer statement discards the unwanted decimal places. Finally, dividing by 100 reverts to the correct level of precision.

Notice that the script in Field 3 still relies on the calculation performed in Field 2. If you do not want to display the intermediate calculation, you can make Field 2 a hidden field.

To round to three decimal places, you can multiply and divide by 1000 instead of 100 in the rounding script. Alternatively, if you need to round to one decimal place, multiply and divide by 10 in the rounding script.

Rounding a number to a whole number is slightly different. In Figure 11-9, Field 4 rounds the value in Field 2 up to a whole number. The script in Field 4 is:

```
calculate:
result = $2 + 0.5
answer = integer result
```

Adding 0.5 rounds up to the next whole number, and the integer statement discards the decimal portion of the number.

Working with Currency Fields

Currency fields are stored internally as a whole number of cents. When performing calculations with a Currency field, you must place the result in a Currency field for the result to be displayed properly.

Figure 11-10 shows a calculation of quantity multiplied by price to give a total cost. The script used in Field 4 is:

```
calculate:
answer = $2 * $3
```

Water 6-pack	
Item Name	Water 6-pack
Quantity	15
Price	\$1.99
Total (Text field)	2985
Total (Currency Fi	\$29.85

End ⏪ ⏩ ⏴ ⏵

Figure 11-10: Calculating with a Currency field

In Field 4, a Text field is used to store the result. This causes the result to be displayed as a whole number of cents. However, when the same script is used in Field 5, a Currency field, the result displays correctly as dollars and cents.

Creating a Counter

Sometimes you may want to create a counter field, so that you can give each record a unique sequential number.

In Figure 11-11, the Count # field increments by one each time a new record is created.

Bookmarks		Wrapping paper		Calendars	
Item Name	Bookmarks	Item Name	Wrapping paper	Item Name	Calendars
Box tag #	1	Box tag #	2	Box tag #	3
Quantity	150	Quantity	25	Quantity	84

Figure 11-11: A script in Field 2 increments a counter for each new record.

Two things are necessary to create a counter field in Field 2: The Advanced Field Property of Autodefault has to be set in Field 2, and the following script has to be added:

```
initialize:
answer = answer + 1
```

The Autodefault property places the value from the previous record into the field. The initialize script then adds 1 to increment the counter.

To start the counter with a number other than 1, you need to enter the number in the counter field of the first record that you create. The next record increments the starting number by 1. For example, if you enter the number 1000 in the counter field, the next record will contain 1001.



If a form with a counter field is redistributed to the handheld, the counter will be reset to 1.

Using Scripts with Dates

Date fields, Date & Time fields, and Time fields can be used in calculations.

Only two types of calculations are meaningful with dates:

- ◆ Subtracting two dates or times to calculate the time elapsed
- ◆ Adding a constant to a date or time to calculate a date in the future

Calculating Time Elapsed

In order to perform calculations with dates, the dates are converted to numbers—the number of seconds since 01/01/1904. A date of 10/31/99 converts to a larger number than a date of 10/30/99.

If you subtract two dates or times, the result typically has to be placed into a Text field or a Numeric field. It is not meaningful, for example, to subtract 9am from 11am and place the result in a Time field.

In Figure 11-12, the time elapsed is calculated using the following script in Field 3:

```
calculate:
answer = $2 - $1
```

The screenshot shows a form field with a title bar that says "9:05 pm". Inside the field, there is a table with the following data:

Starting Time	9:05 pm
Ending Time	11:40 pm
Seconds Elapsed	9300
Minutes Elapsed	155
Hours Elapsed	2.5833333

At the bottom of the field, there is an "End" button and four navigation arrows (left, right, double left, double right).

Figure 11-12: Calculating time elapsed

When you place the result of subtracting two dates into a Text field, the answer is displayed in seconds, as seen in Field 3 of Figure 11-12.

In Field 4, the seconds are converted to minutes by a script in Field 4 that divides by 60 in the following script:

```
calculate:
answer = $3 / 60
```

Another script in Field 5 converts seconds to hours by dividing by 3600 in the following script:

```
calculate:
answer = $3 / 3600
```


If you need to display a time elapsed as a number of days, divide the seconds by 86400, because there are $60 \times 60 \times 24 = 86,400$ seconds in a day.

Expressing Time Elapsed in Hours and Minutes

People don't always relate to time expressed as a fraction of hours. For example, saying that 4.75 hours have elapsed is less meaningful than saying that 4 hours and 45 minutes have elapsed.

As shown in the figure that follows, you can create a form that expresses elapsed time in the form of hours and minutes.

Cleaning vents	
Task name	Cleaning vents
Date Task Perfor	9/28/99
Start Time	9:00 am
End Time	11:35 am
Hours Part	2
Minutes Part	35
Time Elapsed	2 hrs and 35 mins...

End ◀ ◁ ▷ ▶

Displaying elapsed time as hours and minutes (Field 7)

In Field 5, the hours part of the time elapsed is calculated in the following script:

```
calculate:
result = $4 - $3
result = result / 3600
answer = integer result
```

The statement `result = $4 - $3` calculates the elapsed time. The result is in seconds; it is then divided by 3600 to convert to hours. The answer statement then stores just the whole number of hours in Field 5.

In Field 6, the minutes part of the time elapsed is calculated with the following script:

```
calculate:
result = $4 - $3
result = result % 3600
result = result / 60
result = result + 0.5
answer = integer result
```

Continued

Expressing Time Elapsed in Hours and Minutes (Continued)

As in Field 5, the first statement in the script calculates the elapsed time in seconds. The modulo (%) operator in the statement `result = result % 3600` calculates the remainder of seconds after discarding the whole number of hours. The remainder of seconds is then converted to minutes by dividing by 60. The number of minutes is rounded up to the next whole minute by adding 0.5, and then the answer stores just the whole number of minutes.

Finally, to display the time elapsed as the phrase "X hrs and Y mins," the concatenate (&) operator is used in a script in Field 7:

```
calculate:
result = $5 & " hrs and "
result = result & $6
answer = result & " mins"
```

For a really elegant solution, you can make Field 5 and Field 6 hidden fields, so that the handheld user sees only the phrase in Field 7.

Calculating Ages

Calculating a person's current age is a unique form of time elapsed, because it is not just a straightforward number of days and hours – there are years and leap years to consider.

Figure 11-13 shows a form in which the handheld user enters a person's birth date, and a script calculates an estimate of the person's current age.

Liam	
First Name	Liam
Last Name	Jefferson
Date of Birth	9/16/59
Present Age	39.843619

End ⏪ ⏩ ⏴ ⏵

Figure 11-13: Calculating a person's current age

Field 4 is a Text Field with the following script to calculate the person's present age:

```
calculate:
result = now - $3
result = result / 86400
answer = result / 365.25
```

The statement `result = now - $3` calculates the time elapsed from the Date of Birth to the current date and time (assuming that the Palm organizer is set to the correct date and time). This result is expressed in seconds. To convert seconds to days, the script divides the result by 86400. Finally, to convert days to years, it divides by 365.25. Considering that there is a leap year every four years, there are approximately 365.25 days in a year.

The age calculation is only an approximation, because it does not take into account precisely how many leap years have occurred since the person's date of birth.

Adding a Constant to a Date

Apart from subtracting two dates to calculate time elapsed, the only other meaningful calculation that you can perform with dates is to add a constant to a date.

You may want to add a constant to a date if, for example, you need to take a date that the handheld user has entered and figure out what the date would be 14 days into the future.

If you are adding a constant to a date, the constant has to be added in seconds. The result can be placed in a date field. For instance, if you add 86400 to a Date Only field and store the result in another Date Only field, the result will be one day into the future from the first date.

In Figure 11-14, a script is used to estimate the completion date of a project. Given a start date and the estimated length of the project in days, the script in Field 4 calculates the estimated completion date.

Whitman	
Customer Name	Whitman
Project Start Date	7/21/99
Length in days	14
Completion Date	8/4/99

End ⏪ ⏩ ⏴ ⏵

Figure 11-14: Adding a constant to a Date field

The script in Field 4 is as follows:

```
calculate:  
result = $3 * 86400  
answer = $2 + result
```

Adding a constant to a Date field is somewhat limited, because there is no way to account for weekends and holidays. If a project ends two business days from today, and today is Friday, it is not sufficient for a script to add two days to today's date. In a case like this it is better not to use a script at all but to let the handheld user estimate a completion date using the normal calendar in a Date field.

Branching Scripts

Scripts can be used to enable branching on a form. *Branching* means that when the handheld user makes a selection in one field, the response determines which field(s) are displayed next.

Branching scripts work only in Field View – that is, viewing one field at a time on the handheld. Record View does not work with branching, because the Record View screen does not update to jump a user to a field or to show or hide fields.

A branching script should typically be written in a selection field such as a Yes/No field or a Popup List. This is because it is easy to write a script that contains an action for every possible choice that the handheld user makes. A Text field is less predictable – what happens if the handheld user does not type the text that you intend to use to determine where to branch? A Lookup List can be used with a branching script only if there are no more than a few items in the list.

Use `exit` events and `select` events to create branching scripts. A `select` event runs when the handheld user makes a selection in a field. An `exit` event runs when the handheld user exits a field.



Note that `exit` events are more foolproof for writing branching scripts than `select` events. This is because a `select` event will not be triggered if the handheld user fails to make a selection.

The conditional statement, `if . . . then . . . endif`, is used to test the handheld user's response in a field. The `goto` statement is the means by which the form branches to a particular field. The `hide` and `show` statements can be used to hide fields that the user does not need to fill in and display fields that the user has to complete.

Branching with a Jump Popup Field

Some forms use a Jump Popup field to enable the handheld user to select a section of a form to fill in. A Jump Popup field gives the user the ability to jump to a Section field on a form. Once the user has completed the last question in that section, however, a branching script is required if you want to automatically return the user to the Jump Popup field to make another selection.



If you installed Pendragon Forms from the CD-ROM, open the Pendragon Forms Manager and select the Samples category. The form Shopping Survey contains an example of branching from a Jump Popup field.

Figure 11-15 shows a form with a Jump Popup field. Making a selection jumps you to that section of the form. In the last field of each section, the following script is added to return you to the Jump Popup field (Field 2):

```
exit:
goto 2
```

Figure 11-15: A branching script at the end of each section is used to return to a Jump Popup field.

To break out of the loop of always returning to the Jump Popup field, at least one selection should not branch back to the Jump Popup. In Figure 11-15, the Exit section does not branch back, so that if the handheld user selects Exit, the form can be exited.

Branching from a Popup List or Lookup List

When branching from a Popup List or Lookup List, the key thing to remember is that your script needs to cover every possible choice that the handheld user can make, including the case when the user does not make a selection.

To reduce the number of user selections that you have to account for, it is best to use Popup Lists or Lookup Lists with a small number of items in the list.



If you installed Pendragon Forms from the CD-ROM, open the Pendragon Forms Manager and select the Samples category. Distribute the form Breakfast Survey to your handheld, for an example of branching from a Popup List and branching from a Yes/No Checkbox.

Figure 11-16 shows a form that branches when the handheld user makes a selection in a Popup List.

Figure 11-16: Branching from a Popup List

In the Popup List field, the script looks like:

```

exit:
if answer # Cereal then
  show from 5 to 8
  goto 5
endif

if answer # "Hot Breakfast" then
  show from 9 to 10
  goto 9
endif

if answer # Fruit then

```

```

show 11
goto 11
endif

goto 13

```

In the case where the handheld user does not make a selection, the script branches to Field 13.

Note that in the form shown in Figure 11-16, most of the fields have the Advanced Field Property of Hidden set. In the script, the handheld user's selection then determines which fields are made visible via the Show statement.

Sometimes it is not necessary to branch for each item in a Popup List. In Figure 11-17, the form has only two branches – one if the handheld user selects Other, the other if the user selects anything else.

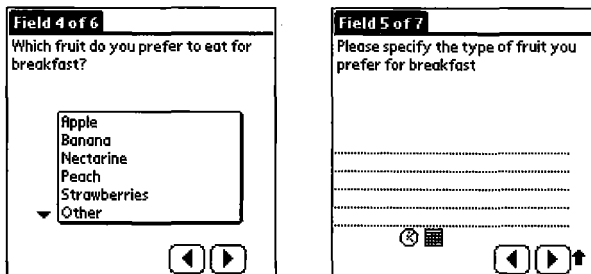


Figure 11-17: Selecting the option Other displays an extra field to fill in.

The script is as follows:

```

exit:
if answer # Other then
  show 12
  goto 12
endif

goto 13

```

Field 12 is made hidden by setting the Advanced Field Property of Hidden. If the handheld user selects Other, Field 12 is made visible for the user to complete. All other selections in the Popup List leave Field 12 hidden and cause a branch to Field 13. (Note that the script refers to the actual field number, in this case Field 12. On the handheld, however, as seen in Figure 11-17, the number that the handheld user sees is field 4, because this is the fourth visible field that the user has seen.)

Branching from a Yes/No Checkbox Field

A Yes/No Checkbox field actually has three possible options: selecting yes, selecting no, or not making a selection at all. A script that branches from a Yes/No field has to account for all three possibilities. In some cases, you may want to make a non-selection have the same effect as choosing No. In other cases, you may prefer a separate branch if no selection is made.

When creating a branch from a Yes/No field, the thing to remember is that internally, a Yes response is stored as the capital letter Y, and a No response is stored as the capital letter N.

In Figure 11-18, a form asks a Yes/No question and then branches depending on the handheld user's response.

Figure 11-18: Branching from a Yes/No Checkbox field

The script in the Yes/No Checkbox field is:

```
exit:
if answer # Y then
  goto 4
endif

show 3
goto 3
```

In this example, responding No or leaving the field blank causes a branch to Field 3.

Scripting Button Fields

Button fields can function only if a `click` event script is written in the field. The `click` events run both in Field View and in Record View.

To create the label on a button, type a word in the Default Value field in the Advanced Field Properties screen. This label can be changed in the script, through the use of the following answer statement:

```
answer = "Name-of-label"
```

Changing the label gives the handheld user the sense that a change occurred as a result of his or her tapping the button. When you create a new record, the label on the button will be reset to the Default Value.

Creating a Button to Lookup from the Address Book

A useful button that you can create is one that enables you to perform a lookup from the built-in Address Book. You can select a person from the Address Book and copy their information into your form.



If you installed Pendragon Forms from the CD-ROM, open the Pendragon Forms Manager and select the Samples category. Distribute the form Address Book Lookup to your handheld, for an example of a Button field that looks up a name and address.

Figure 11-19 shows a form that contains a button to look up from the Address Book. When a contact is selected from the Address Book, the information is copied into the form.

9/15/99	
Date of Visit:	9/15/99
Select customer	(Select)
First name:	
Last name:	
Title:	
Company:	
Address:	
City:	
State:	
Zip:	
Country:	
End [Navigation icons]	

9/15/99	
Date of Visit:	9/15/99
Select customer	(Select)
First name:	Joely
Last name:	Simonsen
Title:	
Company:	BearPaw Production
Address:	184 W. Antler Park
City:	Buffalo Grove
State:	IL
Zip:	60089
Country:	
End [Navigation icons]	

Figure 11-19: Performing a Lookup to the Address Book

The script in Field 2 is as follows:

```
click:
getaddress
```

The `getaddress` statement copies fields from the Address Book that match fields on the form.



For details on how to name fields on your form in order to work with the Address Book, refer to the `GetAddress` statement in the Action Statements section of this chapter.



Because the first field on a form is typically used as the Display Key when reviewing records, it is not recommended that you make a Button field the first field on a form.

Creating a Calculate Button

Calculate scripts automatically run when any field on a form is updated. Imagine that you are adding three numbers, A, B, and C. When you enter number A, the calculate script adds $A + 0 + 0$; then when you enter number B, the script adds $A + B + 0$. Finally, when you enter number C, the script adds $A + B + C$.

An alternative to performing calculations constantly is to create a Button field that performs the calculation only when the handheld user taps the button. The advantage to this is that the handheld user does not see intermediate calculations, only the final result.

The disadvantage of using a Button field for calculations is that if the handheld user changes a field, the calculated result will not automatically update until the button is tapped again. If you don't want to risk possibility of the handheld user's forgetting to tap the button, stick with a `calculate` script instead of a Button field with a `click` event script.



If you installed Pendragon Forms from the CD-ROM, open the Pendragon Forms Manager and select the Samples category. Distribute the form Vacation Cost Estimate to your handheld, for an example of a Calculate button.

In Figure 11-20, tapping the Calculate button calculates a total. The script in the Button field is:

```
click:
result = $2 + $3
```

```

result = result + $4
result = result + $5
result = result + $6
$9 = result + $7
answer = "RECALC"

```

The statement `$9 = result + $7` places the total into Field 9. The statement `answer = "RECALC"` causes the button to display the word RECALC. This lets the user know that the calculation has been performed at least once, and that changes require a recalculation that is performed if the button is tapped again.

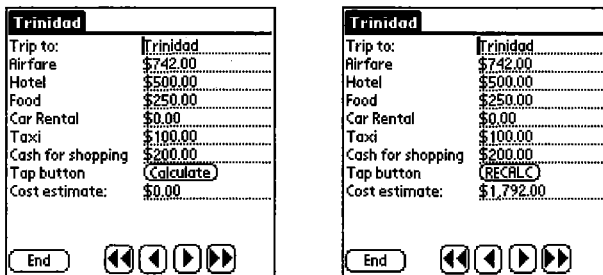


Figure 11-20: Using a Button field to perform a calculation

Creating a Button to Add Records of a Form

If you need to create a total across all records of a form, you can use the `formsum` statement. Adding across all records is processor-intensive, and so a `calculate` script is not recommended, because you do not want to trigger the calculation every time every field on your form changes. Instead, you can create a Button field that performs the calculation only when the button is tapped.



The "Action Statements" topic in this chapter specifies the syntax of the `formsum` statement.

Figure 11-21 shows a form that calculates flight hours for an individual record, and then has a button to calculate the total flight hours logged across all records.



If you installed Pendragon Forms from the CD-ROM, open the Pendragon Forms Manager and select the Samples category. The form Basic Flight Log contains an example of using a Button field to add all the records of a form.

7/22/99	
Flight Date:	7/22/99
Flying From:	3CK
Flying To:	DPA
Type of Plane	C152
Airplane Reg #	5380M
Meter Starting #	321.4
Meter Ending #	321.7
Time Elapsed - Hrs	0.3
Tap here:	(Total)
Total Hrs Logged	89.6

End ◀◀ ◀ ▶▶ ▶▶

Figure 11-21: A click event script can be used with a formsum statement to add across records on a form.

The script in the Button field is:

```
click:
formsum "Basic Flight Log" 8
$10 = result
```

The `formsum` statement looks at the specified form and adds up the value in the specified field (Field 8). The total is placed in the `result` variable. In this script, the total is then placed in Field 10.

The name of the form that you specify in the `formsum` statement can be the same form in which you are writing the script, or it can be a different form.



Adding across records gives the handheld user only a snapshot at a certain point in time. If the user adds, modifies, or deletes one of the records on the handheld, the total will not update unless he or she taps the button again.

Creating a Button to Print a Record

If you want to print directly from the handheld device, you can add a Print button to a form. A Print button gives the handheld user control over when a record is printed.

Printing can be achieved by connecting a serial printer to the handheld serial port. Alternatively, you can print via the handheld infrared port to an infrared printer. Additional infrared software called Bachmann Print Manager, available from www.bachmannsoftware.com, is required to print from the handheld.



An evaluation version of Bachmann Print Manager is available on the CD-ROM. See Appendix C, "What's on the CD-ROM?" for information about this evaluation version.



Refer to Chapter 3, "Entering Data on the Palm Organizer," for more information on printing from the handheld.

Figure 11-22 shows a form with a Button field for printing a record.

Chopper Rescue	
Customer Name	Chopper Rescue.....
Delivery Date/Time	10/29/99 1:19 pm
Item Received	Life rafts.....
Sign here:	Signed.....
Print	(Print)
<input type="button" value="End"/> <input type="button" value="←"/> <input type="button" value="→"/>	

Figure 11-22: A Print button

For infrared printing, the script in the Button field is:

```
click:
print IR
```

For printing to the handheld serial port, the script in the Button field is:

```
click:
print serial
```

Creating a Button to E-Mail a Record

It is possible to create a click event script in a Button field that transmits an e-mail message containing a text version of the current record. MultiMail PRO e-mail software from www.actualsoft.com is required to send an e-mail message that contains a record.



An evaluation version of MultiMail PRO is available on the CD-ROM. See Appendix C, "What's on the CD-ROM," for more information on this evaluation version.

The transmit statement in a script is used to add a text version of a record to the e-mail outbox on the handheld. The e-mail message is sent when next the handheld user connects the handheld device to a modem and sends e-mail using MultiMailPRO.

Sending a record via an e-mail message is not a substitute for performing a HotSync data transfer. If you send records via e-mail, the data is received as a regular e-mail message. It is not possible without significant programming effort to then import data from e-mail messages into a database. Performing a HotSync data transfer directly is a better means of sending data from the handheld to a database.

Figure 11-23 shows a form with an e-mail button.

7/22/99	
Date of Order	7/22/99
Customer #	43846
Customer Name	Sandy Haven Silks
Item 1	Batik scarf
Item 1 Quantity	8
Item 2	Tea tree dress - M
Item 2 Quantity	4
Item 3	Tea tree dress - L
Item 3 Quantity	5
Tap to create e-m	(E-Mail)
<input type="button" value="End"/> <input type="button" value="←"/> <input type="button" value="→"/>	

Figure 11-23: An E-Mail button

The script in the Button field is:

```
click:
transmit multimap "info@abcdef.com"
```

In the preceding script, the specific e-mail recipient is known ahead of time, and so the e-mail address is specified. If you want the handheld user to enter an e-mail address, you can create a field for storing the e-mail address and then reference

that field in the script. For example, if the e-mail address is in Field 7, the script looks like this:

```
click:  
transmit multimap $7
```

Creating a Button to Send a Record to a Website

A variation of the `transmit` statement can be used in a Button field to send a record to a Web address using a Palm VII.



Refer to Chapter 16, "Working with the Palm VII," for information on the `transmit palmnet` statement.

Using a Lookup . . . Within Script

One of the limitations of the regular Lookup List field is that when you make a selection in a list, you can receive only one item of information. For example, if you have a list of items, and you select an item, the item name is displayed with no other information.

Sometimes you may want to receive additional information when you make a selection, such as item name and price, or item name and part number.

A lookup to another form, as described in Chapter 5, "Field Types," can be used to enable you to copy data from one form to another. The limitation of this method is that for a given record, you can perform only one lookup, because data copies over only if the fields in the two forms match. You can't look up a second item without overwriting the first.

An alternative approach may be a `lookup . . . within script`. This type of script enables you select an item from one Lookup List and automatically obtain additional information about that item from a second Lookup List.



If you installed Pendragon Forms from the CD-ROM, open the Pendragon Forms Manager and select the Samples category. The form Simple Order-Taking contains a `lookup . . . within script`.

Figure 11-24 shows a form for taking orders. Tapping in the Item 1 field displays a Lookup List, and making a selection displays the name of the item (Field 3) and, in Field 4, the price of the item.

11/26/99	
Date of order	11/26/99
Customer name	Shirley
Item 1	
Item 1 Price	\$0.00
Item 2	
Item 2 Price	\$0.00
Item 3	
Item 3 Price	\$0.00
Sub-Total:	\$0.00
Tax Rate:	7
Order Total	\$0.00

11/26/99	
Item 1	
Airplane Set	
Astronaut Action Figure	
Baby Doll	
Balloon	
Dinosaurs	
Farm Animal Set	
Marbles	
Toy polar bear	
Toy sea otter	
Train Set	

11/26/99	
Date of order	11/26/99
Customer name	Shirley
Item 1	Toy polar bear
Item 1 Price	\$15.00
Item 2	
Item 2 Price	\$0.00
Item 3	
Item 3 Price	\$0.00
Sub-Total:	\$15.00
Tax Rate:	7
Order Total	16.05

Figure 11-24: A lookup . . . within script enables you to select from two Lookup Lists simultaneously.

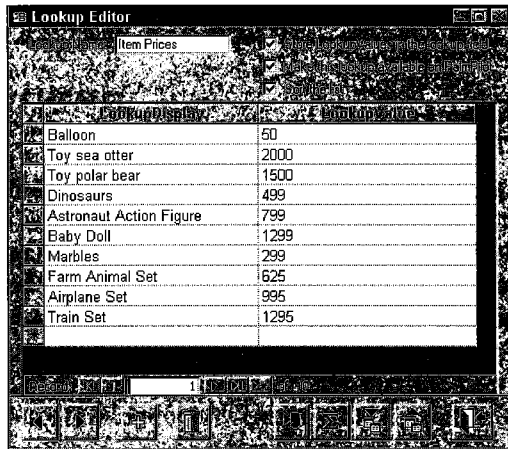
To make it possible to obtain both the item name and the price, you actually need to create two Lookup Lists. Figure 11-25 shows the Lookup List that is used to select the item name. Notice that the Store LookupValues check box in the lookup field is not checked.

LookupDisplay	LookupValue
Balloon	50
Toy sea otter	2000
Toy polar bear	1500
Dinosaurs	499
Astronaut Action Figure	799
Baby Doll	1299
Marbles	299
Farm Animal Set	625
Airplane Set	995
Train Set	1295

Figure 11-25: The Lookup List for selecting an item name does not store Lookup Values.

Figure 11-26 shows the Lookup List that is used to select the prices. This Lookup List has a different name from the first Lookup List, and the Store LookupValues check box in the lookup field is checked. All of the items in the list are identical to those in the first Lookup List.

Notice that prices are listed as a whole number of cents. This is because the price fields on the form are Currency fields, which store dollar amounts as a whole number of cents.



The screenshot shows a window titled 'Lookup Editor' with a tab labeled 'Item Prices'. Inside the window is a table with two columns: 'Item Name' and 'Price'. The table contains the following data:

Item Name	Price
Balloon	60
Toy sea otter	2000
Toy polar bear	1500
Dinosaurs	499
Astronaut Action Figure	799
Baby Doll	1299
Marbles	299
Farm Animal Set	625
Airplane Set	995
Train Set	1295

Figure 11-26: The Lookup List for selecting a price stores LookupValues.

The Item 1 field (Field 3 in Figure 11-24) is a regular Lookup List field that references the Item Names Lookup List. The Item 1 Price (Field 4 in Figure 11-24) is a Currency field with the following script:

```
calculate:
lookup $3 within "Item Prices"
answer = result
```

The script in Field 4 looks up the selected item name from Field 3 in the Lookup List called Item Prices. Because that Lookup List is set to store LookupValues, what gets stored in the result variable is the price of the item. The answer = result statement places the price into Field 4.

Other pairs of fields on the form can reference the same two Lookup Lists. In Figure 11-24, the Item 2 field (Field 5) is also a Lookup List field that references the Item Names Lookup List, and the Item 2 Price field (Field 6) is a Currency field with the following script:

```
calculate:
lookup $5 within "Item Prices"
answer = result
```



The primary thing to remember when using a `lookup . . . within script` is that you have not one but two Lookup Lists to maintain. If you add an item in one Lookup List, you have to add the identical item to the second Lookup List, maintaining case sensitivity.

Using a Validate Script

A `validate` event runs when you exit a record. Typically, a `validate` event is used to verify that the handheld user has entered a unique or correct value in a field.

A `validate` script can be used to display a message to the handheld user to flag any problem fields, such as a duplicate value in a field that is supposed to be unique for every record. Figure 11-27 shows the type of message that can be generated as the user exits the record.



Figure 11-27: A `Validate` script can display a message to the handheld user about problem fields.

To check the uniqueness of a field and display a message if the field is not unique, the `validate` script looks like this:

```
validate:
count 2
if result > 0 then
  msgbox "This record has a duplicate account number."
endif
```

In the preceding example, Field 2 is being tested for a unique value in every record. The `count` statement compares the value of Field 2 in the current record with the value of Field 2 in all other records. The number of records that match the current record is placed in the `result` variable. If there is at least one match, the message is displayed.

The `validate` script just shown does not actually prevent the handheld user from exiting the record. The script merely warns the user that a duplicate record exists.

If you want to prevent the handheld user from exiting the record until the duplicate value has been corrected, you can use the `invalidate` statement. The script in Field 2 can be changed to:

```
validate:
count 2
if result > 0 then
  invalidate "This record has a duplicate account number. "
Endif
```

To reduce the number of scripts on a form, don't use a `validate` script if you can achieve validation of a field via Advanced Field Properties. For instance, setting the Primary Key property ensures that a field must be unique for every record. The Required property ensures that the user does not leave a field blank, and the Max & Min property ensures that a number is entered within a valid numeric range.



For information on making a field a Primary Key or a Required field, see Chapter 6, "Advanced Field Properties."

Repeating Forms

A repeating form is a form that creates a new record as soon as you end one record.

To make a form a repeating form, you can set the Advanced Form Property of AutoRepeat This Form.



For information on the AutoRepeat form property, see Chapter 8, "Advanced Form Properties."

A special case of a repeating form is a form such as a room-by-room inventory, in which you want to enter a room number once for all of the items in the room. To achieve this without scripting, you can choose to make the room location field an Autodefaut field. Each new record that you create inherits the same room location

as the previous record, thus saving time on reentering this information for every record.



For information on the Autodefault field property, see Chapter 6, "Advanced Field Properties."

One limitation of using Autodefault as a solution is that the handheld user has to be aware that when he or she goes to a new room, the room location field has to be changed.

If you want the handheld user to enter the room location information once per room and then not see that field until the next room, you can use scripts to hide the room location field.



If you installed Pendragon Forms from the CD-ROM, open the Pendragon Forms Manager and select the Samples category. The form Room-by-Room Inventory is the form described in Figure 11-28.

Figure 11-28 shows a form that can be used for taking inventory. The first three fields on the form are filled in once per room, and the last two fields are filled in once per item. The first three fields are set to Autodefault (Advanced Field property), and the form is set to AutoRepeat (Advanced Form property). A script in Field 4 is used to hide the first three fields once they have been filled in for a given room.

The script in Field 4 is:

```
enter:
hide from 1 to 3

exit:
if answer = null then
  show from 1 to 3
  $1 = null
  $2 = null
  $3 = null
  abortform
endif

endform
```

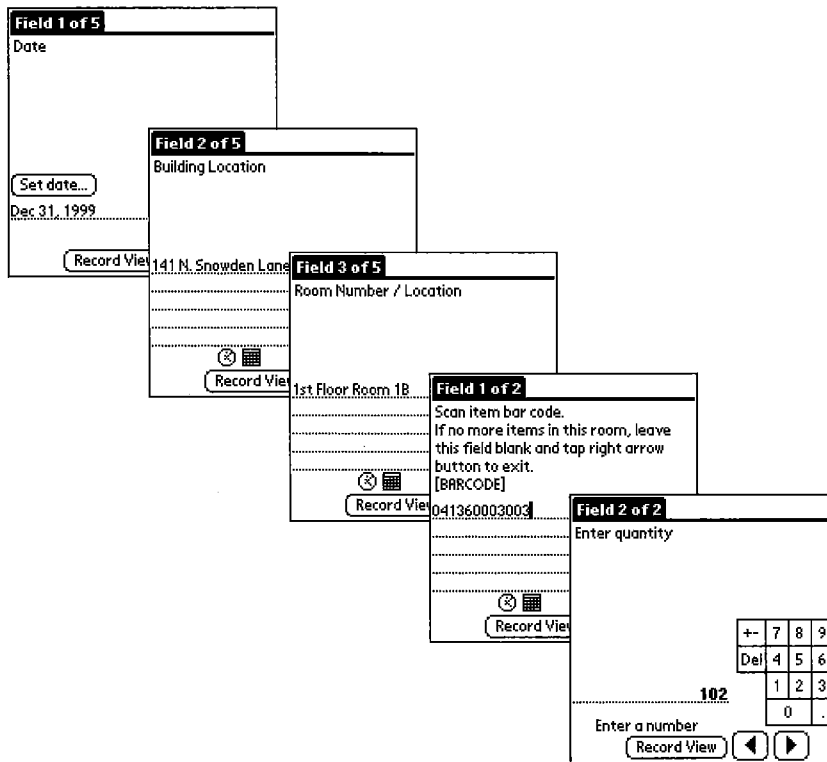


Figure 11-28: A repeating form that hides the first three fields

The first time that the handheld user creates a record, the first three fields on the form are filled in. As the user enters Field 4, the script hides the first three fields. The user can then fill in the bar code and quantity of an item.

When the handheld user exits the first record, the Autorepeat form property creates a new record. The first three fields on the form have the Autodefaut property set, so these fields are already filled in. The fields are also still hidden, and so for the second item in the room, the handheld user needs to enter only the bar code and quantity.

After completing the inventory of the entire room, to end the form the user needs to leave the bar code field blank and tap the Next button (right arrow button). This triggers the `exit` event in the script. The `exit` event displays the first three fields on the form, sets them to null values, and then aborts the blank record and ends the form.

When the next record is created, the first three fields on the form are once again visible, so that the user can fill in the details for the second room.

Printing Scripts

If you need to print your form design in order to see your scripts, you can print a Detailed Form Design Report.

1. In the Pendragon Forms Manager, click the name of the form.
2. Click the File → Detailed Form Design Report.
3. A print preview window appears. Press Ctrl+P to print the report.



The Detailed Form Design Report displays only two fields per page, including scripts. If you have a form with a lot of fields, you can reduce the amount of paper that is used by setting your printer properties to print two pages per physical page.

Summary

This chapter illustrated how scripts enable you to add calculations and branching to a form. Scripts are also used to program Button fields, so that the handheld user can perform various actions by tapping a button.

For faster performance on the handheld, scripts should be avoided where you can use Advanced Field Properties instead.

Chapter 12

Using Bar Codes

IN THIS CHAPTER

- ◆ Using bar code scanners
- ◆ Creating forms that accept bar code input
- ◆ Controlling bar code input
- ◆ Creating bar code applications

ONE OF THE EMERGING USES of the Palm OS family of handheld devices is in conjunction with bar code scanners. With Pendragon Forms, you can create a field on a form that accepts bar code input.

The types of form designs that you can create with a bar code field include:

- ◆ Taking inventory by scanning the product bar code of each item being counted
- ◆ Locating records sent from the PC to the handheld by scanning a bar code and displaying the matching record
- ◆ Creating a library check-in and check-out

Table 12-1 shows the types of bar code scanners that are supported by Pendragon Forms.

TABLE 12-1 BAR CODE SCANNERS

Product Name	Description	How Bar Code Scanner Connects to Palm Device	Manufacturer
SPT 1500 SPT 1700	The SPT 1500 is a Palm III with a built-in bar code scanner. The SPT 1700 is a rugged version of the SPT 1500.	Built-in	Symbol Technologies (www.symbol.com)

Continued

265

TABLE 12-1 BAR CODE SCANNERS (Continued)

Product Name	Description	How Bar Code Scanner Connects to Palm Device	Manufacturer
Copilot	Bar code power supply that clips onto the serial port of a Palm device. You can then connect a bar code wand or laser scanner to the Copilot.	Serial port (clip-on connection)	Dynasys Technologies (www.dyna-sys.com)
Model 4000	Bar code power supply to which you can connect a bar code wand or laser scanner. Connects to Palm device using the Palm modem cable and a 9-pin Male to 25-pin Female adapter.	Serial port (connection via cable)	American Microsystems Ltd. (www.amltd.com)

Can Pendragon Forms Work with Any Bar Code Scanner?

If you do not wish to use any of the bar code scanners listed in Table 12-1, Pendragon Forms can work with any bar code scanner that is self-powered, can transmit data at 9600-N-8-1, and is attached to the serial port of a Palm device.

9600-N-8-1 means 9600 baud, no parity, 8 data bits, 1 stop bit.

The bar code scanner needs to be self-powered because the handheld batteries cannot sustain powering an external device for a long period of time.

The bar code scanner should be configured to terminate data with a carriage return. Pendragon Forms uses the carriage return at the end of a bar code as a signal to power down the handheld serial port and jump to the next field on the form.

Creating a Form to Accept Bar Code Data

In Pendragon Forms, Text fields are the only field types that can accept input from a bar code scanner. You can choose to create a form with one or more Text fields for receiving bar code input.

Pendragon Forms works slightly differently on handheld devices with a built-in bar code scanner than with bar code scanners that are attached to the serial port.

Using an SPT 1500 or SPT 1700

With the SPT 1500 and SPT 1700, you can scan a bar code into a Text field in Field View or in Record View on the handheld.

If you are designing a form that will be used in Field View, you may want add the code [BARCODE] to the end of the field name. If [BARCODE] appears at the end of the field name, as shown in Figure 12-1, and AutoNavigate is switched on, you will automatically jump to the next field on the form after a bar code has been scanned into the current Text field. This speeds up data entry by saving the handheld user from having to tap the Next button to move to the next field.

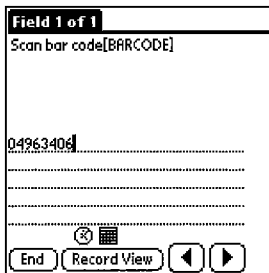


Figure 12-1: Adding [BARCODE] to the end of a field name makes the bar code field function with AutoNavigate.



For information on AutoNavigate, see Chapter 3, "Entering Data on the Palm Organizer."



If you don't want the handheld user to see the code word [BARCODE], insert blank lines (carriage returns) in the field name until [BARCODE] scrolls out of view.

To activate the laser scanner and scan a bar code, press one of the green buttons at the top of the SPT 1500 or SPT 1700 device. The handheld device will beep when the scan is successful.

If you are using Record View for data entry, the cursor must be positioned in the Text field for the bar code to be entered when you activate the laser scanner.

When you view a form in Record View, the cursor is actually positioned in the first Text, Numeric, or Currency field to occur on the form. If you want the form to be capable of receiving bar code input as soon as a new record is created, put the bar code Text field before any other Text, Numeric, or Currency fields on your form. In Figure 12-2, the bar code field is the first field on the form.

# 15 - 041360003003	
1	Scan bar code [B] 041360003003
2	Item Description Batteries
3	Item Price \$3.99
4	Quantity in stock 5
5	Total Value \$19.95

End ⏪ ⏩ ⏴ ⏵

Figure 12-2: For bar code data entry in Record View, put the bar code Text field before any other Text, Numeric, or Currency fields on the form.



If you scan a second time into a bar code field, the second bar code replaces the first.

Using a Bar Code Scanner Attached to the Serial Port

With bar code scanners that are attached to the handheld serial port, Pendragon Forms needs to activate the serial port to detect if a bar code has been scanned.

To automatically activate the serial port when the user enters the bar code field, data must be entered in Field View only. The code word [BARCODE] at the end of the field name identifies to the program that a Text field is a bar code field that requires activation of the serial port. [BARCODE] also causes the field to work with AutoNavigate, as described with the built-in bar code scanners.

The Copilot device can detect when the handheld serial port has been activated and switches on the bar code scanner. You can then scan the bar code into the Text field. With the Model 4000 device, you need to switch on the power supply of the device before you start entering data into the form.

Keeping the handheld serial port activated causes a drain on the handheld batteries. Therefore, Pendragon Forms deactivates the serial port as soon as the bar code has been scanned, or if no bar code is scanned into the field for 45 seconds, whichever comes first.

If the serial port deactivates before you have had a chance to scan the bar code, a message will be displayed as shown in Figure 12-3. You can choose to reactivate the bar code scanner by tapping the Reactivate button.

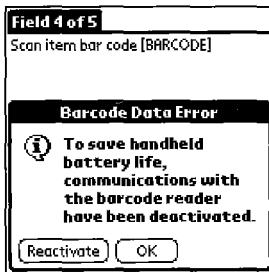


Figure 12-3: With serial port bar code scanners, the serial port is deactivated if there is no activity on the port for 45 seconds.

It is not possible to automatically activate the serial port in Record View. If you need to scan a bar code into a Text field in Record View, you can position the cursor in the Text field and then manually activate the serial port by tapping the handheld Menu button and then tapping the option Acquire Barcode on the Edit menu. (You can also use the Graffiti shortcut /B.)

There are two additional reasons why you may need to manually activate the handheld serial port:

- ◆ In Field View, a bar code field activates the serial port only when you enter the field. If you tap the Previous button to go back to a bar code field, however, the serial port will not activate.
- ◆ If a Text field does not have the code word [BARCODE] to automatically activate the serial port, you can manually activate the serial port.

Controlling Which Bar Codes Can Be Scanned

There are various bar code standards, called bar code symbologies. For instance, bar codes on grocery products often use a 10-character UPCA symbology.

The manuals that ship with the bar code scanners provide examples of the different bar code symbologies.

If you are using a specific bar code symbology, you can configure the bar code scanner to allow only bar codes of that symbology to be scanned.

On the SPT 1500 and SPT 1700, a built-in Diagnostics application on the handheld enables you to select the bar code symbologies that you want to allow. To run the Diagnostics application, tap the handheld Applications button, and then tap the Diag icon. Select the Bar Code Scanner option, and then tap the down arrow to view screen 2 of the Diagnostics application. A list of check boxes enables you to select each bar code symbology that you want to allow.

With the Copilot and Model 4000 products, the manual that comes with the bar code wand or laser gun contains instructions for controlling which bar code symbologies may be scanned. By scanning certain bar codes in the product manual, you can enable or disable various symbologies and set options such as whether prefix and suffix characters in a bar code are scanned.

Using Scripts to Control Bar Code Input on the SPT 1500 and SPT 1700

Symbol Technologies, manufacturer of the SPT 1500 and SPT 1700 devices, provides a programming interface that enables fine control of the bar coding engine. Pendragon Forms utilizes this interface to provide you with the ability to create scripts that control which bar codes can be scanned.

Setting the scanner options in a script is equivalent to setting the scanner options in the built-in Diagnostics application, as described in the previous topic.



Configuring the bar code scanner in a script is a global setting. If you switch off the scanner in a script, for example, or if you enable only one bar code symbology, these settings will affect all fields and all forms, and possibly other applications on the handheld (unless the other applications also make global settings).

Three scripting event procedures work best with scripts that control the bar code scanner:

- ◆ The `open` event lets you make bar code scanner settings every time you create or view a record, in Field View or Record View.
- ◆ The `enter` event, which works only in Field View, lets you set scanner options as you enter a field.
- ◆ The `exit` event lets you set scanner options as you leave a field. The scanner options do not apply to the current field, only to subsequent fields.

Other scripting event procedures are less meaningful with bar code scanner settings. For example, a `calculate` script that runs every time a field is changed is not an appropriate choice of event because you do not need to configure the bar code scanner each time you fill in a field. An `initialize` event that runs only when a new record is created would not allow you to configure the scanner if you wanted to modify an existing record.

The following action statements can be used in scripts to configure the bar code scanner:

- ◆ `{enable | disable} barcode`: Enables or disables the bar code scanner. This is equivalent to switching on or off the laser scanner via the Scanner Enabled check box in the Diagnostics application. If you want to prevent the handheld user from scanning a bar code into a particular Text field, you can use this command. However, because scanner settings are global, you have to remember to switch on the scanner again when needed. When you enable the laser scanner, all the currently selected bar code symbologies can be scanned once more. The following script switches off the laser scanner as the handheld enters the field:

```
enter:  
disable barcode
```

This script switches the laser scanner on:

```
enter:  
enable barcode
```

- ◆ `scanner {enable | disable} all`: Enables or disables all bar code symbologies. If all symbologies are enabled, you can scan any bar code into a Text field. If all symbologies are disabled, no bar codes can be scanned into a Text field. You will also lose the history of which symbologies were previously allowed. To enable symbologies, you can either enable all or enable each desired one in turn. The following script enables all symbologies:

```
enter:  
scanner enable all
```

- ◆ `scanner {enable | disable} symbology`: Allows you to switch on or off the ability to scan a specific bar code symbology. Bar code symbologies that can be referenced are:

```
CODE39
UPCA
UPCE
EAN13
EAN8
D25
I20F5
CODABAR
CODE93
TRIOPTICS39
UCC_EAN128
MSI_PLESSEY
UPCE1
BOOKLAND_EAN
ISBT128
COUPON
```

The following script enables the UPCA symbology:

```
enter:
scanner enable UPCA
```

- ◆ `scanner {system_character | no_preamble} symbology`: Certain bar codes use a leading or prefix character in the bar code. By default, Pendragon Forms allows the prefix character to be recorded when a bar code is scanned.

- The `no_preamble` option switches off the ability to record the prefix character.
- The `system_character` option switches on the ability to record the prefix character.

Symbologies that use a leading character are:

```
UPCA
UPCE
UPCE1
```

The following script switches off the leading character of UPCA bar codes:

```
enter:
scanner no_preamble UPCA
```

- ◆ `scanner {checkdigit | no_checkdigit} symbology`: Certain bar codes use a trailing or suffix character in the bar code. By default, Pendragon Forms allows the suffix character to be recorded when a bar code is scanned.
 - The `no_checkdigit` option switches off the ability to record the suffix character.
 - The `checkdigit` option switches on the ability to record the suffix character.

Symbologies that use a trailing character are:

```
UPCA
UPCE
UPCE1
CODE39
I20F5
MSI_PLESSEY
```

The following script switches off the trailing character of UPCE bar codes:

```
enter:
scanner no_checkdigit UPCE
```

- ◆ `scanner conversion-code {enable | disable}`: Enables or disables bar code conversions. Certain bar codes are abbreviated versions of other bar codes and can therefore be converted to the longer format. For example, a short (8 character) UPCE bar code can be converted to a standard 12-character UPCA bar code. To ensure that all bar codes that are scanned are the same length, you may want to convert the abbreviated format into the longer format. Alternatively, you may want to disallow conversions if you want to record the bar code exactly as it appears on a label.

Valid conversion codes are:

```
UPCETOUPCA
UPCE1TOUPCA
CODE39TOCODE32
EAN8TOEAN13
I20F5TOEAN13
```

A script that enables the conversion of UPCE codes to UPCA is:

```
enter:
scanner UPCETOUPCA enable
```

Customizing Bar Code Input

If you need to scan a particular type of bar code into a field, you may want to configure the bar code scanner for just that field, and then reset the bar code scanner. An `enter` event enables you to configure the bar code scanner as the user enters the field, and an `exit` event enables you to reset the scanner as the user leaves the field.

In Figure 12-4, the script in the bar code field disables all bar code symbologies except UPCE. When the user exits the field, all bar code symbologies are reactivated.

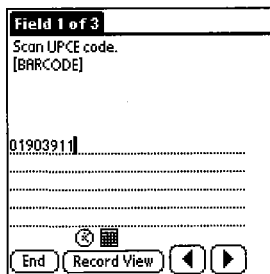


Figure 12-4: An individual bar code field can be configured to accept a specific type of bar code.

The script in the bar code field is:

```
enter:
scanner disable all
scanner enable UPCE

exit:
scanner enable all
```

Figure 12-5 shows a bar code field that automatically converts UPCE to UPCA bar codes. The automatic conversion is deactivated once the user exits the field.

The conversion script is as follows:

```
enter:
scanner UPCETOUPCA enable

exit:
scanner UPCETOUPCA disable
```


Figure 12-5: A script can convert a scanned UPCE bar code (8 characters or fewer) to a 12-digit UPCA bar code.

Sometimes you may need to configure the bar code scanner not just for a single field, but for the entire form. An open script ensures that the bar code scanner is set appropriately every time the handheld user accesses any record.

Figure 12-6 shows a form that does not record the prefix and suffix characters of UPCA bar codes in any bar code fields.

Figure 12-6: An open event script can configure the bar code scanner for all fields on a form.

To configure the bar code scanner not to record prefix and suffix characters of UPCA bar codes, use the following script:

```
open:
scanner no_preamble UPCA
scanner no_checkdigit UPCA
```

As long as no other script changes the bar code settings, all fields on the form will be affected by this script. If the user leaves the Pendragon Forms application to go to another application that configures the bar code scanner differently, the open script will configure the bar code scanner when the user returns to the Pendragon Forms application.

Using Bar Code Fields for Inventory Forms

Bar codes speed up data entry, because it is easier to scan a bar code than to enter a number. If your product inventory is bar coded, you can create several forms for inventory purposes.

The simplest form that can be used for taking inventory is a one-field form. The handheld user can then scan each item individually.

Figure 12-7 shows a one-field form for scanning bar codes. To reduce the time it takes to create a new record, the form can be set to AutoRepeat, meaning that as soon as one record ends, a new record is created. If AutoNavigate is switched on, the [BARCODE] code word in the field name causes the form to move to the next field. Because there are no other fields on the form, AutoNavigate effectively ends the form. The AutoRepeat feature then creates a new record.



If you installed Pendragon Forms from the CD-ROM, open the Pendragon Forms Manager and select the Samples category. The form One Field Scan is a one-field form that autorepeats.

Figure 12-7: A one-field form can be used to scan many bar codes by creating a new record for each bar code scanned.



For more information on the AutoRepeat form property, see Chapter 8, "Advanced Form Properties."

Another type of inventory form enables you to record not just the bar code but also the quantity of items with that bar code.

Figure 12-8 shows a form that contains a field to enter a quantity after the bar code has been scanned. If AutoNavigate is switched on, the [BARCODE] code in the first field will cause the form to jump to the next field as soon as a bar code has been scanned. In Field 2, however, there is no automatic way to detect that the handheld user has completed entering a number in the quantity field. Therefore, the user has to tap the Next button (right arrow button) to end the form. The Advanced Form Property of AutoRepeat can be used to automatically create a new record once the current record has ended.

Figure 12-8: An inventory form can record a bar code and a quantity.

In some inventory applications, you may need to record the location where the item is stored, as well as the quantity in stock.

Typically, many items are stored in the same location, such as a given room in a building. To minimize the need to enter the same location for many items, you can make the form a repeating form.



See the Repeating Forms topic in Chapter 11, "Using Scripts."

Retrieving a Record by Scanning a Bar Code

In some inventory and other applications, you may want to populate the handheld with a list of records. When the handheld user scans a bar code, the matching record can then be displayed and modified.

If you are using an SPT 1500 or SPT 1700 device, Pendragon Forms can accept bar code input on the review screen that displays your list of existing records.

When the handheld user selects the form and taps the Review button, a list of records is displayed. If the user scans a bar code, the bar code data will be entered into the search criteria field, as shown in Figure 12-9.

The figure consists of two screenshots of a handheld application interface. The left screenshot shows a list of records under the heading "Simple Inventory*". The records are listed with their bar codes: 2262747827, 04963406, 00748124, 5000396001136, 1596517265, 01903911, 4136000300, and 7385209650. Below the list, there is a search criteria field with the value "04963406" and a "Done" button. The right screenshot shows a single record with the bar code "04963406". The record has two fields: "Item bar code [B]" with the value "04963406" and "Quantity" with the value "24". Below the record, there is an "End" button and four navigation arrows (left, right, left, right).

Figure 12-9: To find a record that contains a bar code, you can scan the bar code into the search criteria field.

If more than one record contains the bar code, a filtered list of records will be displayed, and the handheld user can make a selection. If only one record matches the bar code, that record will be displayed automatically, as shown on the right in Figure 12-9.



The number of records that you can send to the handheld depends on the size of your form. Refer to Chapter 4, "Planning a Form Design," for information on the memory usage of a record.

Making a Bar Code Field Unique

If you want each record of your form to contain a unique bar code, you can choose to make the bar code field the Primary Key on your form.



For information on Primary Keys, see Chapter 6, "Advanced Field Properties."

If you are not able to make the bar code field the Primary Key, you can use a validate script to check that the value entered into the field is unique.



Information on validate events can be found in Chapter 11, "Using Scripts."

Library Check-In/Check-Out

Pendragon Forms can be used to create a form that tracks whether items – files, books, videos, and so on – have been borrowed or returned. If your items have bar codes, as many books do, you can use the bar codes in your form.

The method that you choose to implement a library check-in/check-out solution depends on the number of items that you have in the library and the amount of information that you want to store on each item.

If the number of items that you have in your library is small enough to store a record for each item on the handheld, then your check-in/check-out solution can be as easy as retrieving and updating the record on the handheld. If each record has a unique bar code, you can retrieve a record by tapping the name of the form and then tapping the Review button, and scanning the bar code in the search criteria field on the review screen.



Refer to the topics "Retrieving a Record by Scanning a Bar Code" and "Making a Bar Code Field Unique" in this chapter. For information on the memory limits of the handheld, see Chapter 4, "Planning a Form Design."

Figure 12-10 shows the type of form that might be used to record whether an item has been borrowed or returned.

9781861871183	
1	Scan item Bar Co 9781861871183
2	Item Description A Little Book for
3	Type of Item *Book
4	Copy # 1
5	Date 9/28/98
6	Status *OUT
7	Last Borrowed b Sarah Smyth

End ◀◀ ◀ ▶▶ ▶▶

Figure 12-10: This type of form tracks borrowed and returned items if the handheld has enough memory to store a record for each item.

If you have too many items to store a record on the handheld for each item, then you cannot use the record retrieval method just described for updating the borrowed or returned status of an item.

In this case, you can create a form that allows you to scan the bar code of the item and record whether the item is being borrowed or returned. Instead of retrieving an existing record, a new record is created on the handheld for each item scanned. Figure 12-11 shows the type of form that can be used. The difference between Figure 12-11 and Figure 12-10 is that in Figure 12-10, the record already exists on the handheld and may include additional fields such as the item name. In Figure 12-11, however, a new record is being created, and a minimum amount of data is recorded so as to save the handheld user data entry time.

9781861871183		
1	Scan bar code	9781861871183
2	Date	10/10/98
3	Status	↓Checked In
4	Borrowed by	Sarah Smyth.....

End ⏪ ⏩ ⏴ ⏵

Figure 12-11: This type of form tracks borrowed and returned items if there are too many items to store on the handheld.

As a precaution against an incorrect new record overwriting a valid existing record, Pendragon Forms does not by default allow a new record on the handheld to update an existing record in the database. The library check-in/check-out case is the exception, because you want the newly scanned record on the handheld to update the borrowed status of the item in the database.

To allow new records on the handheld to overwrite the existing records, set the Advanced Form Property of Allow Inserts on Handheld to Update Existing Records.



If your form maintains one record per item, you will need to ensure that a given item is not stored more than once on the handheld. This is because you can't guarantee the order in which records will be written to the database. You would not, for example, want a record that a book has been checked out yesterday to overwrite a record that the book has been returned today. If you make the bar code of the item the Primary Key, you can ensure that the handheld stores only one record per item.



For information on the Allow Inserts on Handheld to Update Existing Records property, see Chapter 8, "Advanced Form Properties."

A library check-in/check-out solution that uses multiple handheld devices is more complex than the examples shown here.



For information on the issues involved in sharing records across multiple handhelds, see Chapter 15, "Planning a Multi-User Installation."

Summary

This chapter showed you how to create forms that can accept bar code input and outlined the types of bar code scanners that can be used with Pendragon Forms. The chapter looked at methods of controlling which bar codes are scanned into a form and gave examples of applications that can be created with bar code forms.

Part V

Using Pendragon Forms with Existing Databases

IN THIS PART

CHAPTER 13

Linking to an External Access Database

CHAPTER 14

Linking to an ODBC Database

Chapter 13

Linking to an External Access Database

IN THIS CHAPTER

- ◆ Creating a form from an external Access database table
- ◆ Linking a form to an external database table
- ◆ Linking a parent form and subform to related database tables
- ◆ Linking a form to a database query

ONE OF THE NEW FEATURES in Pendragon Forms version 3.0 is the capability to link to an external Microsoft Access database. This means that if you created your own Access database prior to using Pendragon Forms, you can use Pendragon Forms to send data from your database to your Palm device. You can also set up a link so that records that you create or modify on the handheld update your Access database directly.

The steps involved in creating a Pendragon form that links to an external Access database are:

1. Preparing your existing database table for use with Pendragon Forms
2. Creating a form from an existing database table
3. Editing the form design for use on the handheld
4. Controlling which records go to the handheld
5. Linking the form to the external database table
6. Sending the form to the handheld
7. Viewing data on the PC

To illustrate these steps, we will look at a sample database for recording customer information, and we will use Pendragon Forms to send the information in this database to the handheld.

When linking a Pendragon form to an existing Access database table, you will need to keep in mind the following questions:

- ◆ Are handheld users allowed to create new records, or is the data from the PC for reference only on the handheld?
- ◆ Will each handheld user receive separate records, or do all handheld users share records?
- ◆ How will you limit the number of records that are sent to the handheld?

Your answers to these questions will influence how you set up your Pendragon form to link to your Access database table.

Preparing Your Database Table for Use with Pendragon Forms

Before you begin the process of linking a Pendragon Form to an external Access database table, you may want to look at your database table to see whether you need to make modifications to it in order for it to work with Pendragon Forms.



One of the first things to identify is whether any changes can be made to your database table. If other programs, Access queries, or reports are relying on your database table as it currently exists, you may not be able to make changes to the database table at all.

When Pendragon Forms attempts to update a table with changes made on the handheld, it first looks to see if a corresponding record exists in the table. To identify the corresponding record, it compares the Primary Key fields of the records from the handheld with the primary keys of records in the table, looking for a match. If your external Access table does not have a primary key, you will not be able to update existing records, though you will be able to view and add new data on the handheld. If records are removed from the handheld after synchronization, a primary key is probably not important. However, if records will be maintained on the handheld (for instance, via a Completion Checkbox) and be updatable on the handheld, you will need a primary key for your Access table.

If you decide to add a primary key to your Access database table, you will need to select a field or combination of fields to be the primary key. The fields you choose should be unique for each record in the table and each record likely to be added to the table. For instance, if your database table stores customer information, and each customer has a unique account number, the account number can be selected as the primary key. If you have a patient list, you might choose the Social Security number as the primary key. If you are recording patient visits, and the

same patient can make repeated visits, you might make Social Security number and Date of Visit the primary key.

If no suitable combination of fields can be found in the table, you can add a column to the table to act as your primary key. Microsoft Access provides an Autonumber type that generates a unique sequential number for a database column. The AutoNumber type is often used for unique numbers such as a customer number or order number. However, creating an additional column for the primary key should not be the first route you take, if there are preexisting columns that can serve as a unique identifier.

Figure 13-1 shows an Access database table that has been created to store a list of customers. Each customer has a unique account number that is generated via the use of an AutoNumber field in Access. The AutoNumber field is made the primary key in the database table.

Primary Key field

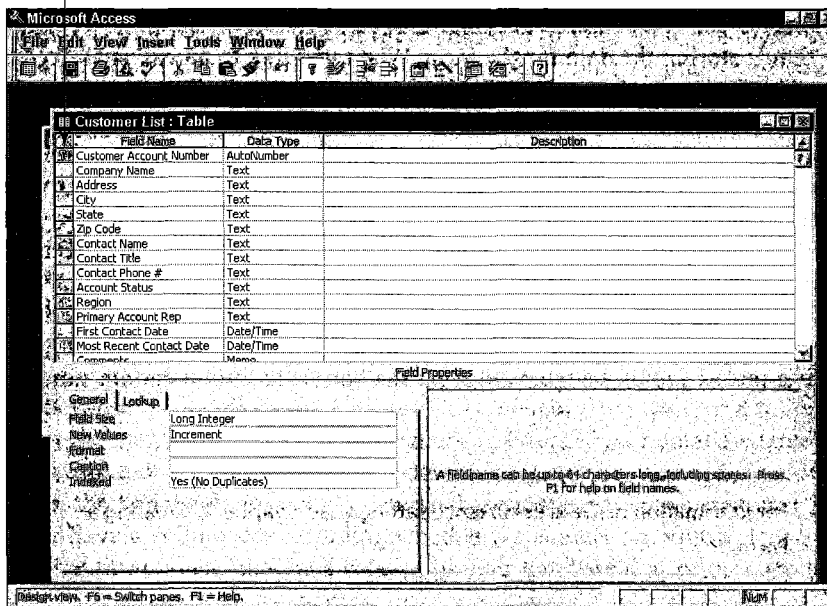


Figure 13-1: An Access database table for customer information

In a typical application created in Microsoft Access, an Access form is also created to enable the user to enter data into the database table. Figure 13-2 shows the type of form that might be created in Access to act as the front-end for user input.

Figure 13-2: A form created in Access to enable user input on the PC

Pendragon Forms generates four additional database columns whenever a form is frozen: RecordID, UnitID, UserName, and TimeStamp. When data for a Pendragon form is being stored in the Pendragon Forms Manager database (Forms3.mdb, or Forms32k.mdb if you have Access 2000), these four fields are visible in the database table that is storing the data for the form. When you link to an external Access database table, however, these four fields do not automatically exist in your database table.

The UserName and TimeStamp fields are significant in Pendragon Forms. (RecordID and UnitID are present only for backward compatibility with earlier versions of Pendragon Forms.)

The UserName field stores the handheld user name, and by default, only records with a given UserName are sent to the handheld with the same name.

The TimeStamp field stores the creation date and time of a record. Pendragon Forms uses the TimeStamp field in combination with Data Persistence settings to determine if a record should be removed from the handheld after a certain number of days.



For information on Data Persistence settings, see Chapter 2, "Creating a Form."

You will have to decide whether you want to add the UserName and TimeStamp columns to your existing database table. If you add these columns, named exactly as in Pendragon Forms, you will need to prepopulate the existing records in the database with values in the UserName and TimeStamp columns. New records created on the handheld will automatically have values for these fields.

If your database table already has a field that is equivalent to the `UserName` field, you do not need to add a `UserName` field. For example, in Figure 13-2, the database table already has a field that records which sales representative is assigned to each customer. If you wanted to give each handheld user only the records assigned to that person, you could use the `Primary Account Rep` field to achieve the same function as the `UserName` field.



If you decide to use another field as a substitute for the `UserName` field, remember that the names you enter into this field must exactly match the handheld user names. For data entry on the PC, you may want to create a drop-down list so that the PC user selects the appropriate handheld name instead of having to enter text.

Adding the `UserName` and `TimeStamp` columns to your database table is optional. If you prefer not to add these columns to your database table, you can use the `Additional Download Criteria` to determine which records from your database table are sent to the handheld, and when the records should be removed from the handheld.



Refer to the topic "Controlling Which Records Go to the Handheld" in this chapter for information on `Additional Download Criteria`.

Because the handheld device has limited memory, you will need to have a way to limit the number of records on the handheld. Check if your database table already has a field or fields that can be used for determining whether a record should be removed from the handheld. For example, a database table that stores customer information may have a column for recording whether a customer account is active or inactive. Alternatively, there may be a column in your database that records the last transaction date of a customer, and you can use this field in your `Additional Download Criteria` to place only records in the last 90 days on the handheld, for example.

If you want to use a `Completion Checkbox` field on your Pendragon Form, so that when the handheld user checks the box the record is marked for removal from the Palm organizer, you can add a column in your database table for storing the data from the `Completion Checkbox` field.

Once you have added any fields that you need to add to your database table, the next step is to use the database table as the basis for creating a Pendragon form.

Creating a Form from an Existing Database Table

To create a form from an external Access database table, click the Import button in the Pendragon Forms Manager. As shown in Figure 13-3, a Select Import Source window appears. Click the Import Access Table Design button.

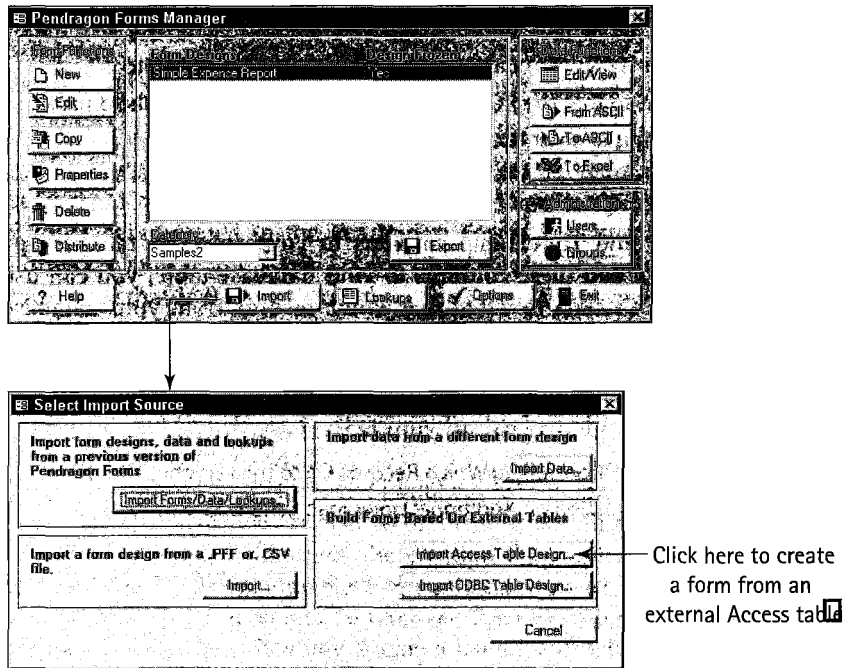


Figure 13-3: The Import button in the Forms Manager window enables you to build a form design based on an external Access database table.

A Select Access Database window is displayed. As shown in Figure 13-4, you can select the Access database (.mdb file) that contains the database table that you want to use as the basis for a Pendragon form.

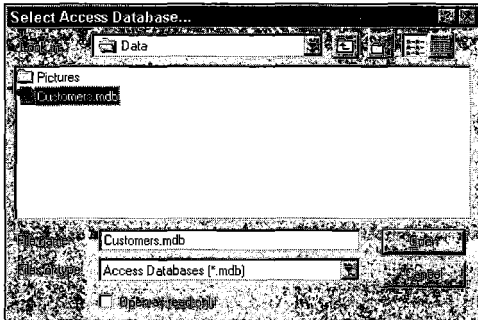


Figure 13-4: Select the Access database with which you want to work.

Once you have selected a database to work with, click the Open button in the Select Access Database window. A Table Selector window is displayed. As shown in Figure 13-5, the Table Selector window displays a list of all the database tables and queries in the selected database. Choose which database table you want to use to create a Pendragon form.

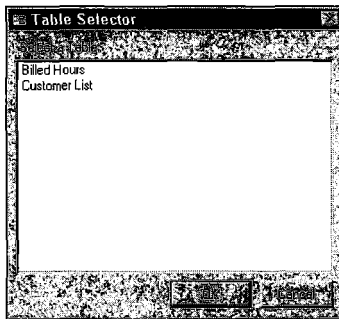


Figure 13-5: Select the database table that you want to use as the basis for a Pendragon form.



You can link to a query instead of a database table. See the topic “Linking to an External Database Query” in this chapter.

After you select a database table and click OK, an Import Data dialog box appears. As shown in Figure 13-6, the Import Data dialog box prompts you to type a name for the Pendragon form that you are creating.

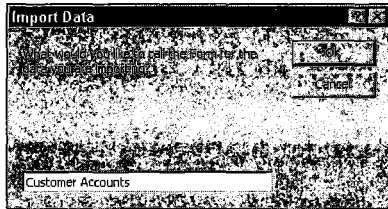


Figure 13-6: Type a name for the form that will be used on the handheld.

Now that a form has been created based on the Access database table, the next step is to edit the form for use on the handheld.

Editing a Form Design for Use on the Handheld

When an Access database table is used as the basis for creating a form, the resulting form design typically needs editing in order to work well on the handheld.

To edit the form, click the name of the form in the Forms Manager, and then click the Edit button.

A form that is created from a database table contains a field for every database column in the external database table. However, if you do not need to send every data field to the handheld, you can choose to delete fields from the form.

The field names that are generated when the form is created are identical to the database column names. In many cases you may want to modify the field names to make them more meaningful to the handheld user. For instance, a field name based on a database column called FNAME is more meaningful if changed to First Name.



Although you can change the field names, do not change the Advanced Field Property of Column Name. The Column Names in the form design need to match those in the external database table in order that the form link to the database table.

For some fields, you can change the field type to facilitate ease of data entry by the handheld user. For example, the Text field in Figure 13-7 has been changed to

a Popup List in Figure 13-8 because there is a limited number of options and it is easier for the handheld user to select an option than to enter freeform text.

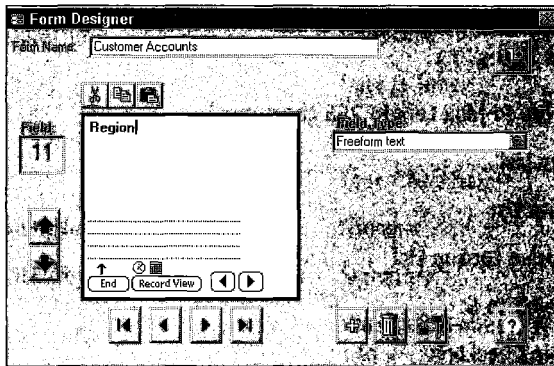


Figure 13-7: You can change the field type of a Text field to a Popup List or Lookup List to make data entry easier for the handheld user.

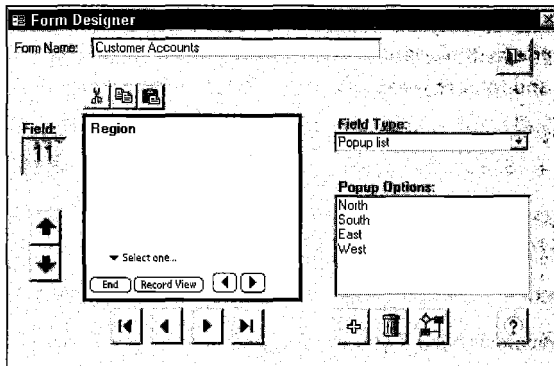


Figure 13-8: A Text field is converted to a Popup List because the possible options for this field (North, South, East and West) are all known in advance.

The field types that you can use in a Pendragon form depend on the field types that you are using in your existing database table. When you create a form from an external database table, Pendragon Forms automatically makes the field types on the form compatible with the column data types in the underlying external database table.

Table 13-1 shows which Pendragon field types can be used with which Access field types.

TABLE 13-1 COMPATIBILITY BETWEEN PENDRAGON FIELD TYPES AND ACCESS FIELD TYPES

Existing Access Database Field Types	Compatible Pendragon Forms Field Types
Text (up to 255 characters)	Text (up to Max Length 255 characters) Popup List Lookup List Exclusive Lookup List Option 1 to 5 Yes/No Checkbox Completion Checkbox
Memo	Text (with Max Length set to 2000) Multi-Selection List
Number: Byte	Numeric (with Advanced Field Properties: Min = 0, Max = 255, and Integer = checked)
Number: Integer	Numeric (with Advanced Field Properties: Min = -16384, Max = 16383, and Integer = checked)
Number: Long Integer	Numeric (with Advanced Field Properties: Min = -2147483648, Max = 2147483647, and Integer = checked)
Number: Single Precision	Numeric with a specific range. Not recommended if you are creating an Access database table from scratch, because the Palm organizer has more precision than a single-precision field in Access. Double precision is preferred.
Number: Double Precision	Numeric
Yes/No	Yes/No Checkbox (with a default value of N or Y, according to the convention used in your existing database table). A default value is required because Access does not accept a null value in a Yes/No field, whereas on the handheld the user can leave a Yes/No Checkbox blank.
Currency	Currency

Existing Access Database Field Types	Compatible Pendragon Forms Field Types
Date/Time	Date Only Date & Time Time Time Checkbox
AutoNumber	Numeric
OLE Object	Signature Works with bitmap file format only, with an image size of 150 x 150 monochrome pixels.

Another editing change that you can make to your Pendragon form is to add Jump Popup fields, Section fields, Subform fields, or Single Subform fields as necessary to make the form usable on the handheld. These fields do not store data, and so they do not need to link to the external Access database table. However, use of these fields on a Pendragon form may make the form easier to navigate on the handheld.



Chapter 5, "Field Types," contains information about Jump Popup fields, Section fields, Subform fields, and Single Subform fields.

You can also add fields to your form for storing calculations or interim results of calculations that are useful for the handheld user but do not need to be stored in the external database table.

Setting Advanced Field Properties

If you create a Pendragon form from an existing Access database table, Pendragon Forms sets some Advanced Field Properties in order to make the fields on the form compatible with the underlying external database table. Although these Advanced Field Properties can be edited, it is recommended that you do so only if you need to place more restrictions on a field, not fewer. For example, you can make a numeric range smaller, not bigger.

Any Primary Key fields in the external database table will have the Advanced Field Property of Primary Key set.



Primary Key fields are also set as Read-Only for individual records that are sent from the PC to the handheld. This is done to prevent the primary key from accidentally being changed on the handheld. However, new records created on the handheld will have editable Primary Key fields unless you specify otherwise.

Numeric fields in the external database table will have the appropriate Advanced Field Properties of Max, Min, and Integer set, depending on the type of numeric field being used. Table 13-1 describes the possible settings.

Memo fields in the external database table will be set to Text fields with Max Length 2000 characters.



Refer to the sidebar topic “What Happens if the Data in the External Database Is Too Large to Fit on the Handheld?” for the scenario in which a Memo field in the external Access database holds more data than can fit in a 2000-character Text field on the handheld.

AutoNumber fields in the external Access database are a special case in Pendragon Forms. An AutoNumber is a unique number that is automatically generated by Microsoft Access for each record in the external database, and as such, AutoNumber fields cannot be edited or set by users. This means that Pendragon forms cannot store values in AutoNumber fields because attempting to do so would result in a database error.

If your existing database table contains an AutoNumber field, Microsoft Access will generate a sequential number for each record when the records are sent from the handheld to the database. However, if the AutoNumber field is also the Primary Key field, then when the handheld user creates new records on the Palm device, the field will need to be filled with a unique number on the handheld. The unique number generated on the handheld will eventually be discarded and replaced with a number generated by Microsoft Access.

Pendragon Forms converts AutoNumber fields in the existing database table to Numeric fields with certain Advanced Field Properties set, as shown in Figure 13-9.

If the AutoNumber field is a Primary Key field, the Primary Key property is set. In addition, the Do Not Upload to PC property is set, so that any number generated in this field is used only on the handheld, not sent to the PC.



If you do not want the handheld user to change the value in the AutoNumber field, set the Read-Only property.

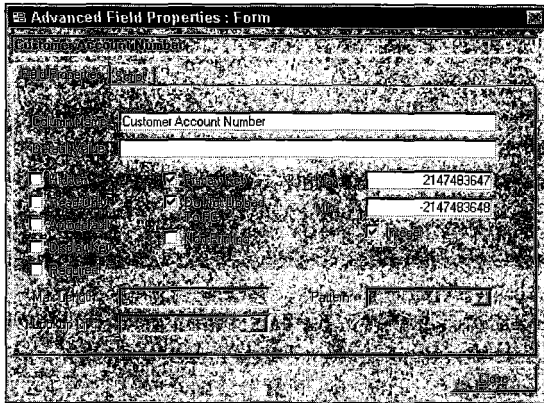


Figure 13-9: Advanced Field Property settings for an AutoNumber field

To generate a unique number on the handheld in the AutoNumber field, Pendragon Forms automatically adds a script to the field when the form is created. Figure 13-10 shows the script that is created.

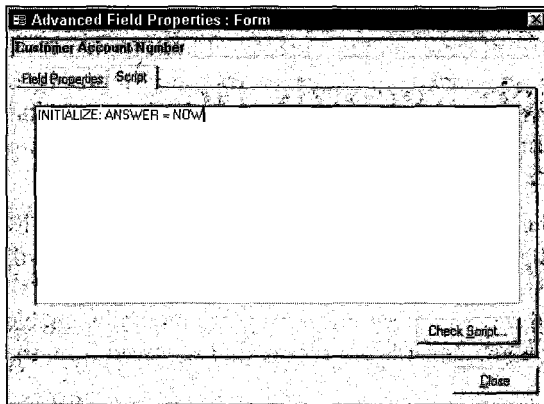


Figure 13-10: The script that Pendragon Forms adds to an AutoNumber field

The initialize script runs only once when each new record is created. The statement `answer = now` assigns the current date and time to the field. Because dates are stored internally as the number of seconds since January 1, 1904, the script essentially places a number into the Numeric field. The number is unique, because two records cannot be created at exactly the same date and time.

What Happens if Data in the External Database Is Too Large to Fit on the Handheld?

A Memo field in Microsoft Access can store as much as 2 gigabytes (2GB) of data. In contrast, the maximum size of a Text field in Pendragon Forms is 2000 characters, or 2 kilobytes.

If a record in the external database table contains a Memo field with more than 2000 characters, only the first 2000 characters will be sent to the handheld. To prevent the 2KB on the handheld from overwriting the larger field in the database during subsequent synchronization, the field is made Read-Only and Non-Updating when stored to the handheld. Only the specific record that is too large will have these Advanced Field Properties set. Other records that do not exceed the 2KB limit will not be made Read-Only or Non-Updating.

Similarly, if a Currency field in the external database is greater than \$10,000,000.00 or less than -\$10,000,000.00, the field will be too large to store on the handheld. In this case the field will display null on the handheld and will also be Read-Only and Non-Updating on the server for the specific records that are affected.

Controlling Which Records Go to the Handheld

Because storage space on the handheld is limited, it is very important to specify the rules that govern when a record will be sent to the handheld and when a record will be removed from the handheld.

The Form Properties window and the Advanced Form Properties window enable you to set up the rules to determine which records are sent to and removed from the handheld.

In the Pendragon Forms Manager, click the name of the form, and then click the Properties button. The Form Properties window, as shown in Figure 13-11, is displayed.

In the Data Persistence section, check the box Keep a Copy of Records on PalmPilot in order to send records from the existing database to the handheld. The Additional Download Criteria in the Advanced Form Properties window can then be used to fine-tune which records are sent to the handheld. This is the typical method of maintaining which records are on the handheld.

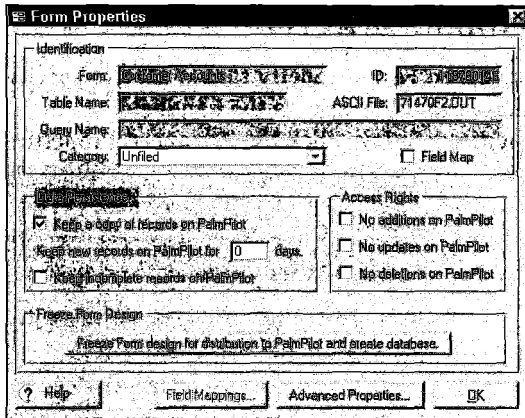


Figure 13-11: The Form Properties window

Alternatively, if you have a TimeStamp column in your external database table, you can choose Keep New Records on PalmPilot for X days. You will need to populate the existing records in your database with a creation TimeStamp. New records that are created on the handheld will have the creation TimeStamp automatically generated on the handheld.

The third Data Persistence option, Keep Incomplete Records on PalmPilot, can be used only if you have added a column in your database table for storing information from a Completion Checkbox field on the form. In that scenario, all records marked as Yes will be removed from the handheld.

If none of these three options is specified, all records will be removed from the handheld during synchronization.

Setting Handheld Access Rights

In the Form Properties window shown in Figure 13-11, the Access Rights section is used to select whether the handheld user can create new records or modify existing records.

Pendragon Forms does not set the Access Rights automatically, so if you do not want to allow the handheld user to add new records on the handheld or to be able to update existing records, set the appropriate Access Rights here.



For information on Access Rights, see the topic "Limiting Handheld Access Rights" in Chapter 3, "Entering Data on the Palm Organizer."

Setting Additional Download Criteria

Additional Download Criteria is an Advanced Form Property that is used to specify which records from the existing database should be sent to the handheld.

To access the Advanced Form Properties window, click the Advanced Properties button in the Form Properties window.



For information on Additional Download Criteria, see Chapter 8, "Advanced Form Properties."

Figure 13-12 shows Additional Download Criteria set up to send only the records assigned to a particular Account Representative to the handheld user of the same name. The value `##USERNAME##` is a wildcard that represents the handheld user name of the device that is synchronizing. If a `UserName` column has been added to the database table to store the handheld user name, the Additional Download Criteria will be:

[UserName] = ##USERNAME##

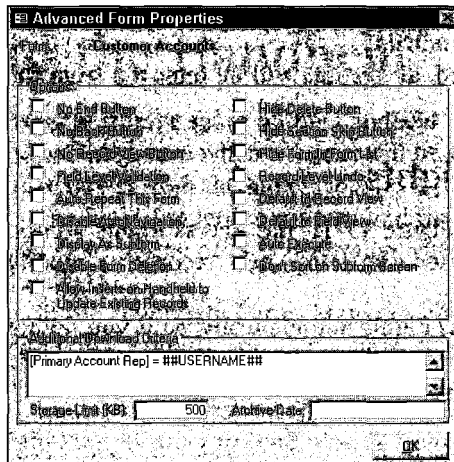


Figure 13-12: Using Additional Download Criteria to send specific records from the database to the matching handheld user name

Sharing records across handheld units is a special case in which the ##USER-NAME## wildcard is not used. If applicable, other download criteria can be used to select which records are sent to the handheld. In Figure 13-13, the handheld user receives all records in which the customer account status is Active.

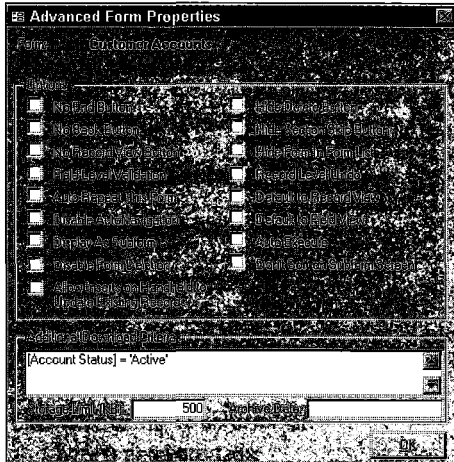


Figure 13-13: Using Additional Download Criteria to send all records with account status Active to the handheld

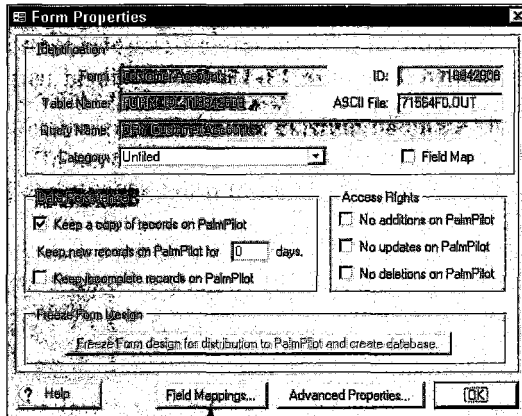


For a look at the issues involved in sharing records across handheld units, see Chapter 15, "Planning a Multi-User Installation."

Linking the Form to the External Database Table

Up to this point, the form that has been created based on the existing Access database table is not actually linked to the database table.

Before you can set up the link to the external database table, you will need to freeze the form design. As shown in Figure 13-14, freezing the form makes the Field Mappings button in the Form Properties window active.



This button becomes active
after you freeze a form

Figure 13-14: The Field Mappings button in the Form Properties window is accessible only after a form has been frozen.

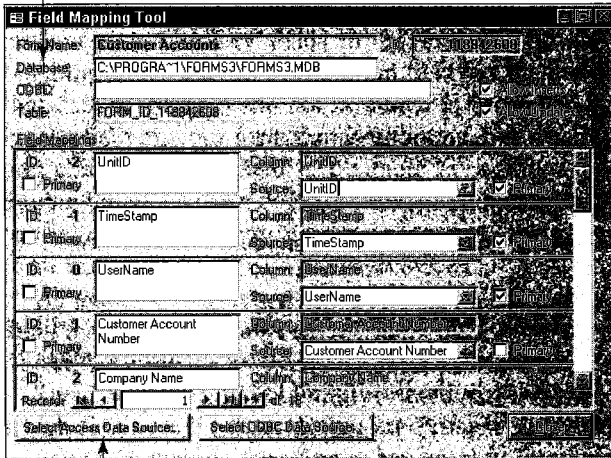


If you change the Additional Download Criteria for your form, you will need to repeat the Field Mapping process before the changes take effect.

Click the Field Mappings button to establish the link to the external database. A Field Mapping Tool appears. As shown in Figure 13-15, the default database to which the form links is the Pendragon Forms database. At this point in time, if you do not establish a link to an external database table, the form will send its data back to the Pendragon Forms database.

To link to an external Access database table, click the Select Access Data Source button. As shown in Figure 13-16, a Select Access Database window appears. Select the Access database file that contains the table you want to link with, and click the Open button.

The default setting is to map the Forms3 database



Click here to link to an external Access database table

Figure 13-15: The Field Mapping Tool window

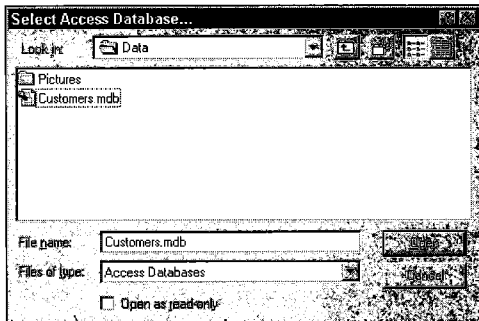


Figure 13-16: Select the external Access database that you want to open.

Once you have selected an Access database to open, a Table Selector window is displayed, as shown in Figure 13-17. Choose the specific database table to which you want to link your form, and then click OK.

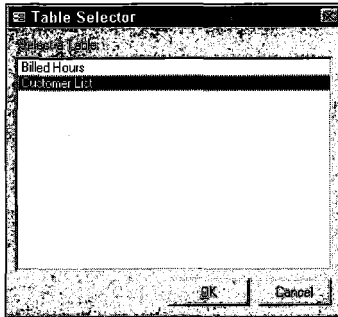


Figure 13-17: Select the database table to which you want to link your Pendragon form.

When a database table has been selected, the Field Mapping Tool window will be displayed once more. As shown in Figure 13-18, the Database field now maps to the external Access database instead of to the Pendragon Forms database.

The external database

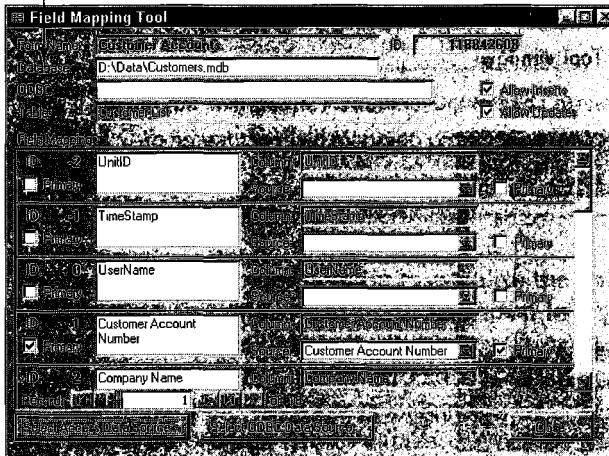


Figure 13-18: The Field Mapping Tool window showing links to an external database table

Once you link to the external database table, the Field Mapping Tool window displays the relationship between the fields of the external database table and the Pendragon form.

Each field on your Pendragon form is listed in the Field Mapping Tool window. The Column field is the database column name for that field. The Source field is the

column from your external database table. If the form was originally created from the database table, the Column fields should be mapped to the correct Source fields of the same names. Scroll down the list of fields to verify this.

Certain fields on the form may be unbound, meaning that the fields on the form are not linked to any fields in the external Access database table. In the Field Mapping Tool window, an unbound field has the Source field left blank. This means that data in the field will not be stored in the external Access database table. If you have a field on your form that is unbound, but which you need to store in the table, you can link it to a field in the table by making a selection in the corresponding Source list box shown in Figure 13-18. This situation may arise if you added fields to the table after importing the design from the external table, or if you created a form from scratch.

As described earlier, primary keys play an important role in synchronization. The Field Mapping Tool will not allow field mapping to proceed if the primary keys on the table do not match the keys on your form design.

The Field Mapping Tool indicates with check boxes which fields on the table and which fields on the form design are Primary Key fields. These check boxes cannot be edited. To change the primary key on your form, use the form designer to set the Advanced Field Property for the relevant fields. To change the primary key on the table, you will need to use the Microsoft Access table designer. After making these changes you can return to the Field Mapping Tool.

If the table does not have a primary key, you will see a warning message when you save the field mappings, and updates to existing records will be disabled.

If you are not using the RecordID, UnitID, UserName, or TimeStamp fields that Pendragon Forms automatically generates, then these four fields can be unbound fields. Similarly, fields such as Jump Popup and Section fields that are used for navigation on the handheld do not need to be bound to the external database table. Fields containing calculations that are used only on the handheld can also be left unbound if the data is not needed in the external database table.

The Field Mapping Tool window also contains two check boxes: Allow Inserts and Allow Updates. Check the Allow Inserts box if you want the handheld user to be able to create new records that will be added to the external database table. Check the Allow Updates box if you want the handheld user to modify records that are sent from the database table to the handheld.



If you choose not to allow either inserts or updates to the external database table, also set the handheld Access Rights in a similar fashion. Otherwise, the handheld user will actually be able to create new records or modify existing records, but then during synchronization these changes will not be allowed to modify the external database table.



For information on Access Rights, see the topic "Limiting Handheld Access Rights" in Chapter 3, "Entering Data on the Palm Organizer."

Once you are satisfied that the fields on the form are mapped correctly to the columns in your external database table, you can close the Field Mapping Tool window. If there are fields on the form that are unbound – that is, the fields do not map to columns in the external database table – you will see a message similar to the one in Figure 13-19.

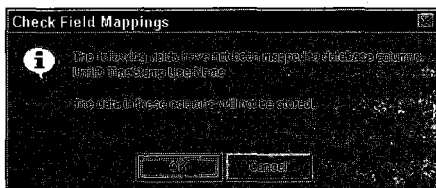


Figure 13-19: Closing the Field Mapping Tool window will display a message to alert you if there are fields on the form that are unbound.

If you do not need any of the unbound fields to be stored in your external database table, click OK. Close the Form Properties window. Your form is now ready to be sent to the handheld.



If you decide that you need to store one of the unbound fields, and there are no appropriate columns in the table, you will need to close the Field Mapping Tool window and then add a column to your database table. You will need to repeat the field mapping process in Pendragon Forms so that the previously unbound field is linked to a column in your external database table.



If you make changes to the Advanced Form Properties of your Pendragon form, you will need to repeat the process of linking to the external database table.

Sending a Form to the Handheld

To send the form to the handheld, click the name of the form in the Forms Manager, and then click the Distribute button.



If you are using User Groups, you will also need to assign the form to a User Group. See Chapter 7, "Synchronization Rules."

Perform a HotSync data transfer to send the form to the handheld. Tap the name of the form, and then tap the Review button to verify that the correct records have been downloaded from the database.

Figure 13-20 shows a form on the Palm device that has been populated with data from an external Access database. The Primary Key field is identified with an icon of a key.

Customer Account #	01
Company Name	Turtle Beach Tours
Address	1850 Sandy Way
City	Highland Park
State	IL
Zip Code	60035
Contact Name	Charlie Gilberte
Contact Title	Marketing VP
Contact Phone #	847-555-3484
Account Status	Active
Region	East

End

Figure 13-20: A form with records populated from an external database

To prevent the accidental corruption of the primary key on the handheld, records that are populated from the database have the Primary Key field(s) set to Read-Only. Because the handheld user cannot change the primary key, the record is guaranteed to match the primary key of the existing record in the database. If the handheld user modifies the record, Pendragon Forms will be able to locate the correct record to update in the database when the handheld user next performs a HotSync data transfer.



If you get HotSync error messages while synchronizing to an external Access database table, refer to Appendix A, "Troubleshooting Tips."

Viewing Data on the PC

If you are linking to an external Access database, you can no longer use the Edit/View button in the Pendragon Forms Manager to view your data. Although Pendragon Forms still creates a database table in the Forms database when you freeze the form, data associated with the form goes directly to your external database table during synchronization.

To view the new or updated records sent from the handheld to the PC, open the external database directly. The database table to which you have linked your form will contain the records from the handheld.

Linking a Parent Form and Subform to External Database Tables

Most Access databases store related information in several tables. This not only reduces the amount of storage space required but also prevents inconsistencies and duplicated information. Suppose your database stores a list of transactions with customers. This information could be stored in a single table that includes customer name, address, phone, and the transaction description. However, because the customer name, address, and phone rarely change, this information is likely to be duplicated on each transaction record. In addition to using up storage space, the address information could be slightly different on each record. When it comes time to create a list of customers (say, for mailing brochures), it may be very difficult to find the duplicate records.

A solution to this type of problem is to create two tables, one for the customers and one for the transactions. The Customers table stores one record for each customer indexed by a customer number. The Transaction table stores the customer number and the transaction details. To print a report for a transaction, information is combined from both tables by matching the customer number on the Customer and Transaction tables. Using multiple tables can make some tasks a little bit more complicated, but doing so can significantly improve the quality of the data that is stored. For example, the database can be instructed not to allow Transaction records to be added without there being a corresponding Customer record. Such constraints are known as referential integrity rules. The ability to combine data from multiple tables and create referential integrity constraints makes Microsoft Access a "relational database."

The Pendragon Forms software that runs on the handheld is not powerful enough to be a truly relational database, but its ability to use subforms to link related records enables it to work as an effective data entry and viewing tool.

To link Pendragon Forms with a Customers/Transactions database, create a parent form that links to the Customers table with customer information, and then create a subform that links to the Transactions table. The parent form should include a Subform List field enabling the handheld user to navigate to subform records.



For information on subforms, refer to Chapter 5, “Field Types.”

This same approach works for many similar applications—for example, Customers/WorkOrders, Patients/Visits, and so on.

To illustrate the process of linking to related tables, we will use a time tracking database as an example. The time tracking database tracks work performed for customers by consultants who will record their hours on Palm organizers.

The database has two tables, Customer List and Billed Hours. The Customer List table stores customer information, while the Billed Hours table contains a record for each time hours are recorded by a consultant. Figure 13-21 shows the two tables and the relationship between them.



For information on how to create relationships between database tables, refer to the documentation that ships with the Microsoft Access product.

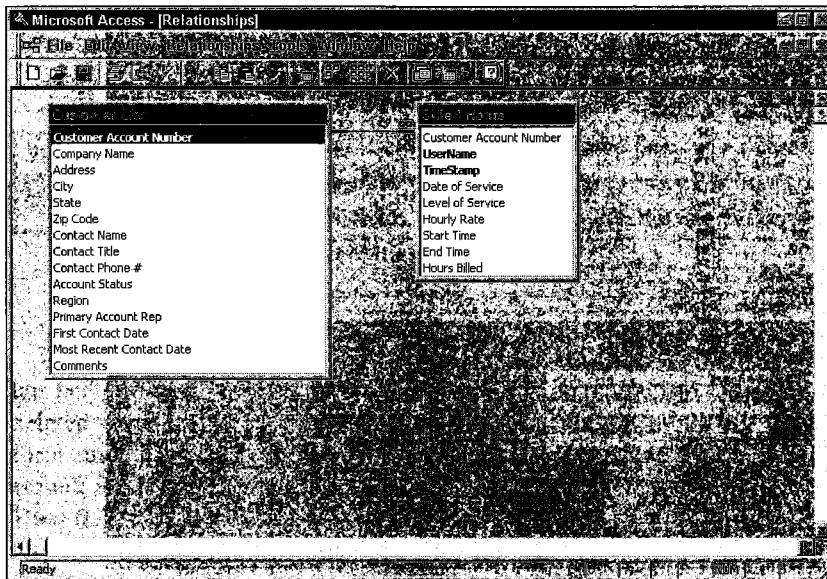


Figure 13-21: A relationship between database tables in the external database

Order of Synchronization

The procedure for creating Pendragon forms to work with these tables is similar to the procedure for a single table. However, the order in which you import the table designs as Pendragon Form designs may be important. If you have referential integrity constraints preventing records from being created in one table without a corresponding record in the other table, you must make certain that records are added to the respective tables in the correct order. Generally, this means that parent forms should synchronize before subforms.

When you create a form in Pendragon Forms, the form is automatically assigned a unique Form ID number that is based on the creation date and time of the form. When you synchronize to an external database, forms with a lower Form ID number will synchronize before those with a higher Form ID number. To have the parent form synchronize first, simply create or import it before the subform.

If the parent form is not created first, it will have a higher Form ID number than the subform, and during synchronization the subform will be synchronized first. This may cause errors if the referential integrity rules require that a parent record must exist before a subform record is created.

Back to our example, Figure 13-22 shows the Customer List table that will be used as the basis for creating the parent form. The database table is imported to create the form, and then the form is edited for use on the handheld.

Field Name	Data Type	Constraint
Customer Account Number	AutoNumber	
Company Name	Text	
Address	Text	
City	Text	
State	Text	
Zip Code	Text	
Contact Name	Text	
Contact Title	Text	
Contact Phone #	Text	
Account Status	Text	
Region	Text	
Primary Account Rep	Text	
First Contact Date	Date/Time	
Most Recent Contact Date	Date/Time	
Comments	Memo	

Figure 13-22: An Access database table that will be used to create a parent form

To create a form design, import the table as described earlier in this chapter.



See the topic "Creating a Form from an Existing Database Table" in this chapter.

During the process of editing the form for use on the handheld, a subform field is added to the form. The subform field provides a link from the parent form to the subform on the handheld. There is no meaningful data in the subform field itself, and so this field will be unbound, meaning that the external database table does not need to contain a column to store data in the subform field.

After the editing process, the criteria for downloading records to the handheld are set up in the Advanced Form Properties window. In this example, the Additional Download Criteria field is set so that only records associated with those customers with "Active" account status will be sent to the handheld.

Once download criteria have been established, the form is frozen and then linked to the external Access database table via the Field Mappings button in the Form Properties window.

Figure 13-23 shows the Access database table that will be used in this example as the basis for creating the subform. In this database table, a UserName and a TimeStamp column have been added to the database table to record the handheld user and the date and time the record was created. This pair of fields is guaranteed to be unique, so they are designated Primary Key fields.



If you are using the UserName and TimeStamp fields as a primary key, you do not need to use other fields as part of the primary key. Conversely, if you are using your own fields as the primary key, do not use the UserName and TimeStamp as part of the primary key.

The subform is then created from the Billed Hours database table. After the form has been created, the form design is edited. The Username and TimeStamp fields are deleted from the form design, because these fields are implicitly a part of every form and the handheld user does not need to see these fields. Other fields are edited; for example, the Start Time and End Time fields are changed from a Date & Time field to a Time field to facilitate entering just the time.

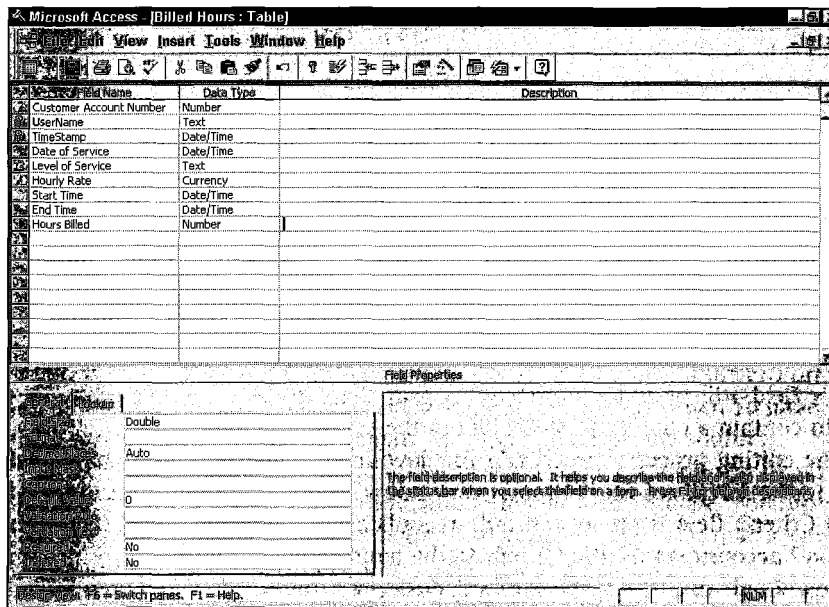


Figure 13-23: An Access database table that will be used to create the subform

After you've edited the subform for use on the handheld, you establish the rules for which records are stored on the handheld. Because the database table is storing the creation TimeStamp field that Pendragon Forms generates, the Data Persistence option of Keep New Records on PalmPilot for X Days can be used to store subform records on the handheld for 30 days, for example.

In the case of several handheld users creating billing records, the handheld users do not need to share billing records. To ensure that each handheld user receives only subform records that he or she created, the Additional Download Criteria field in the Advanced Form Properties window is set to:

```
[UserName] = ##USERNAME##
```

Setting the Additional Download Criteria as shown here works in this case because the UserName field is being stored in the external database table. If handheld users needed to share subform records, different Additional Download Criteria would be required to set the rules for which records are maintained on the handheld.

Once the Data Persistence options and/or the Additional Download Criteria for the subform have been specified, the subform can be frozen and then linked to the external database table via the Field Mappings button in the Form Properties window.

The parent form and the subform can then be distributed to the handheld. On the left of Figure 13-24 you can see the parent form on the handheld. To create a new subform record, tap one of the icons in the subform field (the Work Performed field). A Subform screen, shown in the center of Figure 13-24, displays any existing subform records. To create a new subform record, tap the Add button. A new subform record can then be filled in, as shown on the right of Figure 13-24.

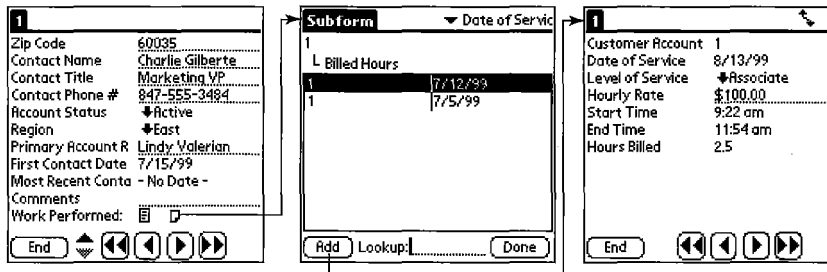


Figure 13-24: Entering a new subform record from the parent form

When you perform a HotSync data transfer, the external database is updated with any new parent and subform records.

Linking to an External Database Query

In addition to being able to link to an external database table, you can also choose to link a Pendragon form to an external database query.

A query is a subset of rows or columns from a table or combination of tables. The simplest queries just present data from a table in a sorted order, or present only those rows from a table that meet some criteria. More advanced queries can create calculated fields or compute sums, totals, and averages.

To a database user, queries appear to be tables in many respects. They are displayed on screen like tables, and many queries are editable just like tables. However, queries do not work the same way on the handheld as database tables do.

The difference between linking to a database table and linking to a query is that whereas you can update records in a database table from the handheld, records in a database query cannot generally be updated on the handheld because there is no primary key. This means that data from the query must be read-only on the handheld. However, if the query itself is updatable, you may be able to append records to the query just as you could append records to a table that has no primary key.

Although records that are sent to the handheld by linking to a database query are read-only on the handheld, linking to a query is still very useful. For example, if you have a query that gives you sales information, you can create a Pendragon form that links to the query to give you up-to-the-minute sales information every time you synchronize.

To link to a database query, follow the procedure for creating a form from an external Access database. Instead of selecting an Access database table, however, choose to link to a query.



See "Creating a Form from an Existing Database Table" in this chapter.

When you import a form design based on an external database, the Table Selector window shown in Figure 13-25 enables you to choose either an external database table or a query to link to.

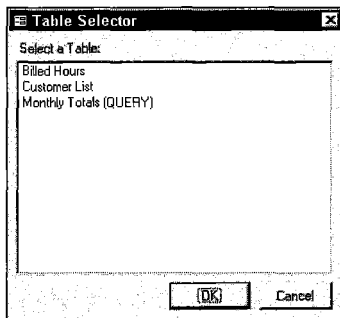


Figure 13-25: You can choose to create a form based on an external database query.

Once the form has been created, you can edit the form design to change field names or delete fields from the form. If the fields will all be read-only, you do not really need to make changes that would make data entry easier on the handheld, such as changing Text fields to Lookup Lists.

Don't forget to set the Access Rights to prevent updates on the handheld. If the query is not updatable at all, then you should also check the No Additions on PalmPilot check box in the Form Properties window.

Freeze the form and then link the form to the external database query by clicking the Field Mappings button in the Form Properties window. When you close the Field Mapping Tool window, a dialog box will remind you that the records on the handheld cannot be updated. Figure 13-26 illustrates this message. Click OK to continue.



Figure 13-26: When linking to a query, you will be reminded that you cannot update records on the handheld.

Distribute the form to the handheld. When you synchronize, the query will run and you will receive the records that match the criteria specified in the query. Figure 13-27 shows a form that links to a query to generate monthly sales by geographic region.

A screenshot of a data entry form. At the top left, there is a header "99-08". Below it is a table with four rows of data:

Month	99-08
Region	East
Total Hours	13.5
Amount Billed	3550

At the bottom of the form, there is an "End" button on the left and four navigation icons (back, forward, first, last) on the right.

Figure 13-27: A form that is linked to a query receives updated records from the query on every HotSync data transfer.

Summary

This chapter outlined the basics of creating a Pendragon form that is based on an external Access database table. Once such a form is created, the form design can be edited to optimize data entry on the handheld. The form can then be linked to the external database table, so that records synchronize directly between the external database and the handheld. Variations on linking to an external Access database table include linking a parent and a subform to related database tables, and linking a form to an Access database query.

Chapter 14

Linking to an ODBC Database

IN THIS CHAPTER

- ◆ Understanding ODBC
- ◆ Linking to ODBC via a linked table in Access
- ◆ Direct mapping to an ODBC database table
- ◆ Addressing security concerns

IN ADDITION TO BEING able to link to external Microsoft Access databases, Pendragon Forms supports linking to many types of external database systems via ODBC.

Understanding ODBC

Open Database Connectivity (ODBC) is a Microsoft standard for accessing databases. Its goal is to enable programmers to write programs that work with any kind of database format – for example, Microsoft Access, Paradox, or Oracle – without having to write specifically for any one database. It's like a universal language for databases.

A good analogy for this can be found in the printer manager in the Windows operating system. When you buy a printer, it usually comes with a disk that includes printer drivers that tell Windows how to print to the specific printer model that you purchased. In fact, Windows itself does not know how to print anything; it simply manages printer drivers and print jobs. The great advantage of this system is that software programs need only know how to tell Windows to print a page; they don't need to know anything about specific printers. For their part, printer manufacturers can feel free to develop new printers that work with preexisting software applications – all they have to do is write a printer driver to go with the new printer.

In a similar way, ODBC itself just provides the framework for the communication between applications and databases. The ODBC driver, like a printer driver, handles the manufacturer-specific details. An application developer can write software to use an ODBC-compliant database and later decide which database will be used to actually store the data.

Different database systems have different capabilities, so not all ODBC databases will be capable of performing the same tasks. For example, there is an ODBC driver to access information in Microsoft Excel spreadsheet files. Because Microsoft Excel is not a true database program, it does not support many database features such as primary keys and unique indexes. An ODBC-compliant application that requires primary keys to operate will not be able to store data in Excel format. This is analogous to a Windows program's supporting only color printers.

ODBC drivers are available for most popular database formats. Some ODBC drivers are provided by Microsoft, some by database vendors, and others by independent companies. Some ODBC drivers are free, and others can be quite expensive. In some cases, there may be several ODBC drivers for the same database system from different vendors offering different levels of performance or capability.

Before you can access a database with ODBC, you must first create an ODBC Data Source for the specific database you wish to access. A Data Source specifies the location of the database, identifies the driver for the database, and assigns a name to the database for use with ODBC applications. Here are a couple of examples:

- ◆ A data source called "Sales" that refers to a SQL Server database at network address 205.109.22.3
- ◆ A data source called "MarketingSurveys" that refers to a Microsoft Access database file at C:\Mktg.mdb

Linking to an ODBC Database

There are two ways to link Pendragon Forms to an ODBC database:

- ◆ By creating a linked table in Access
- ◆ By mapping directly to an ODBC table

As a general rule, use a linked table where possible.

Microsoft Access has an innate capability to work with external database tables by creating linked tables. A linked table behaves like a normal Access table, but instead of storing its data in the Access database file, it stores it in the external database.

With Microsoft Access you can create linked tables that connect to other Microsoft Access database tables, and to dBase, Excel, FoxPro, Paradox, and text files. You can also view and edit data in external databases via ODBC by creating a table linked to an ODBC data source.

When you are linking Pendragon Forms to external databases, a linked table created within the Forms database often provides the best solution. The idea is to

create a linked table with Microsoft Access and then link Pendragon Forms to the linked table by following the directions in Chapter 13.

Here are some reasons why you should consider using linked tables:

- ◆ Linked tables may be the only solution for creating links to certain file formats such as Access, dBase, Excel, FoxPro, Paradox, and text files.
- ◆ Once you have created a linked table, you can verify that the link provides you with the information you need by directly opening and interacting with the linked table within Microsoft Access.
- ◆ Linked tables have good performance and in some cases provide the most flexible way to connect to external databases.



You need to have the full version of Microsoft Access 97 or Access 2000 in order to link to an external ODBC database. You also need to have a working knowledge of ODBC databases.

The steps for linking to an external ODBC database are summarized in Table 14-1.

TABLE 14-1 PROCESS FOR LINKING TO ODBC DATABASES

Step	Linking to dBase, FoxPro, or Excel	Linking to Other ODBC Databases via a Linked Table in Access	Linking to Other ODBC Databases via Mapping Directly to ODBC
1	N/A	Create a machine Data Source.	
2	Create a linked table in Access, and create a Pendragon form based on the linked table.		Map directly to an ODBC table, and create a Pendragon form based on the ODBC table.
3	Edit the form for use on the handheld.		
4	Control which records go to the handheld.		
5	Link the form to the external table.		

Creating a Linked Table in Access

Creating a linked table in the Pendragon Forms database enables you to view your external database table from within the Forms3.mdb database (or Forms32k.mdb if you have Access 2000). Once a linked table has been created, you can create a Pendragon form based on the database table, in the same way that you would create a form based on an external Access database table.

Microsoft Access ships with what are called ISAM drivers for linking to dBase, FoxPro, Paradox, Microsoft Excel, and text (comma-delimited ASCII) files. The ISAM drivers enable you to create a linked table in the Pendragon Forms database without using ODBC.



If you want to link to an external Access database that is password protected, creating a linked table in the Forms database will enable you to synchronize. However, because your password will be stored in the linked table, users with access to the linked table will be able to see the contents of your password-protected database.

Note that although you can create linked tables for Excel and text tables, these file formats are not databases and do not support primary keys. Because Pendragon Forms requires a primary key on the handheld, this means that if you send records from the PC to the handheld, you cannot modify the records on the handheld to update the existing records on the PC. The absence of a primary key in the Excel or text file makes Pendragon Forms unable to identify which records should be updated. Linking to Excel or text should be used primarily if you want to send reference data to the handheld that you do not need to update. You can also link to Excel or text if you are only creating new records on the handheld, not modifying existing records.

You can also create a linked table in Access to link to ODBC databases such as those hosted by SQL Server, Oracle, and Informix. Before you can create a linked table to these types of databases, you will need to create a machine Data Source so that you can reference your external database from within Access. A machine Data Source is a named profile for the ODBC database that is available to all programs on a given PC. (See accompanying sidebar for information on creating a machine Data Source.)

To create a linked table, open the Pendragon Forms Manager and bring the Forms3: Database window to the foreground. (If you are using Access 2000, the database is called Forms32K.) Figure 14-1 illustrates the Forms3: Database window.



If you cannot see the Forms3: Database window, click Window → Cascade.

Creating a Machine Data Source

On the PC, click Start → Settings → Control Panel. In the Control Panel window, double-click the 32bit ODBC icon.

An ODBC Data Source Administrator window will appear. Click the System DSN tab. Check if a Data Source already exists for the database to which you want to link. If a Data Source already exists, you do not need to create one.

If you need to create a Data Source, click the Add button.

Select a driver (for example, an Oracle driver) and enter a name to be used as the Data Source Name. Select the data source – that is, the path and specific database name to which you want to link.

Different database drivers will require different information to create a Data Source. Refer to the Help file that shipped with your ODBC driver, or refer to the manufacturer of the driver for more information.

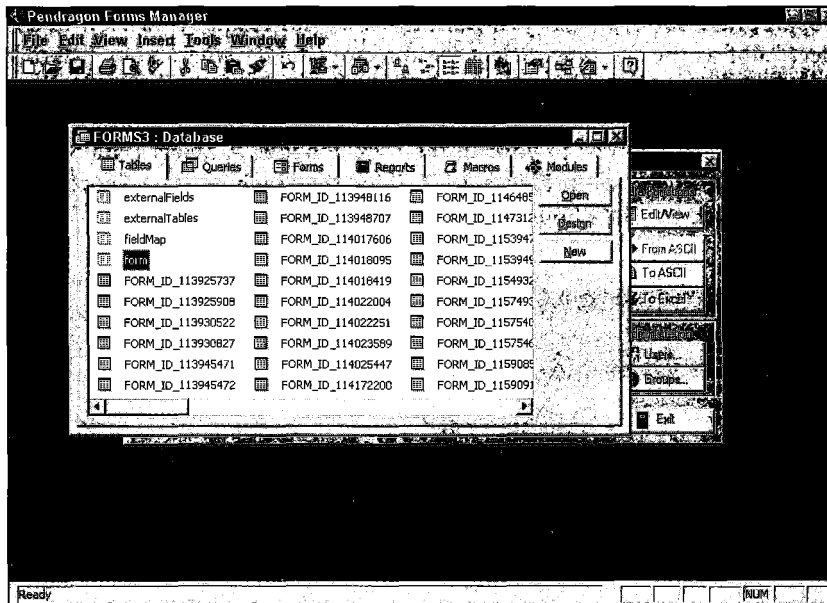
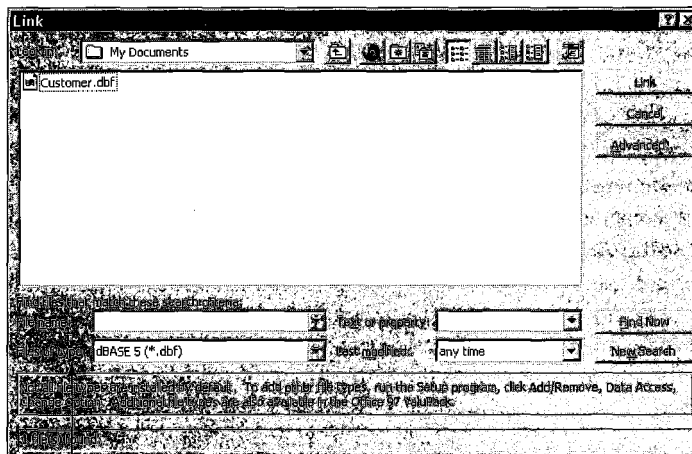


Figure 14-1: Bring the Forms3: Database window to the foreground.

Click File → Get External Data → Link Tables. As shown in Figure 14-2, a Link window will appear.



Select database type

Figure 14-2: The Link window

If you are creating a link to a dBase, FoxPro, Paradox, Excel, or text database, select the appropriate file type that you want to view – for example, to view dBase 5 files, select *.dbf as the file type. Select the directory folder where your external database is located. Choose the file name and click Open.

If you are creating a link to a SQL Server, Oracle, or Informix database, select ODBC databases as the file type to view in the Link window. A Select Data Source window will appear, as shown in Figure 14-3. Click the Machine Data Source tab, and then double-click the Data Source Name that you previously created.

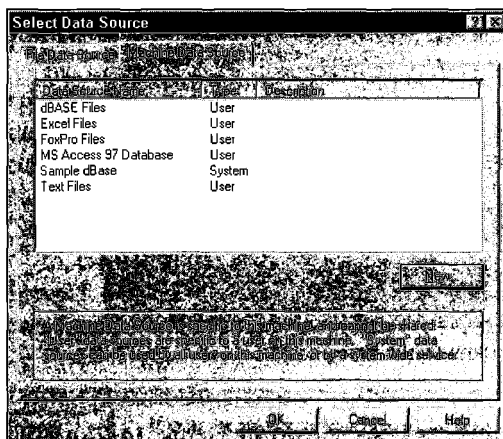


Figure 14-3: The Select Data Source window

If you have to enter a password to access the external database, you will be prompted for the password.

Once you have accessed the external database – by selecting the database name or the machine Data Source Name – a Link Tables window will appear. As shown in Figure 14-4, the Link Tables window displays a list of database tables in the selected database. Click a database table name and click OK.

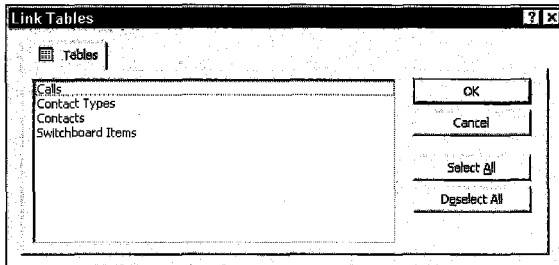


Figure 14-4: The Link Tables window lets you select a database table in the external database.

If your database does not have a primary key but the driver is capable of simulating one, you may be prompted to select fields to act as the primary key fields. The primary key selected will also be used on the handheld.

In the Forms3: Database window, click the Tables tab. By default, the name of the linked table is the same as the name of the table in the external database, so look for the name of the database table to which you just linked. As shown in Figure 14-5, the linked table will contain an arrow next to the table name to indicate that it is a linked table.

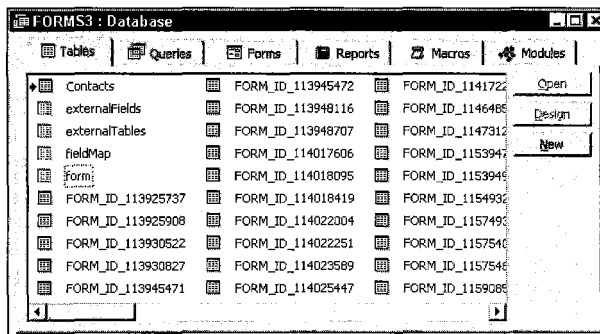


Figure 14-5: In Microsoft Access, linked tables are marked with an arrow.

Click the name of the linked table, and then click the Open button to view the data in the database table. Verify that you are linking to the correct database table and that you can see the records that you want to send to the handheld. Then close the database table window.

Bring the Pendragon Forms window to the foreground. As shown in Figure 14-6, click the Import button, and then click Import Access Table Design.

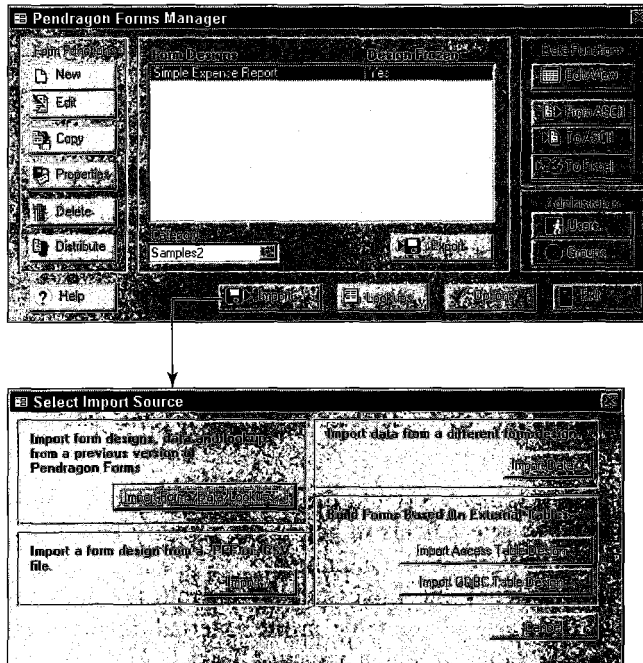


Figure 14-6: Click the Import Access Table Design button to create a Pendragon form from a linked table.

A Select Access database window appears. As shown in Figure 14-7, choose the Forms3.mdb file (or Forms32K.mdb if you have Access 2000) and click Open.

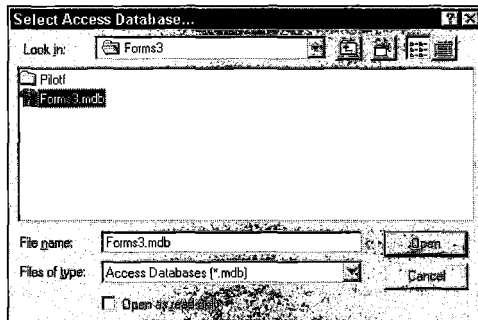


Figure 14-7: Because the linked table was created in the Forms3 database, choose the Forms3 database to open.

A Table Selector window, shown in Figure 14-8, displays the list of database tables and queries in the Forms3 database. Select the name of the linked table.

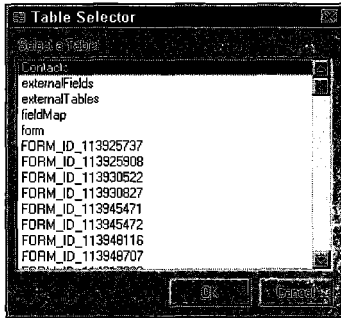


Figure 14-8: The Table Selector window

After you have selected the linked table, you will be prompted for a name for the Pendragon form that you are creating.

Mapping Directly to an ODBC Table

Mapping directly to an ODBC table is a slightly faster process than creating a linked table in Access. However, using a linked table may be faster for synchronizing data during the HotSync process.

Before you can map to ODBC, you will need to create a machine Data Source for your ODBC database.



See the sidebar topic “Creating a Machine Data Source” in this chapter.

To set up a direct link to ODBC, click the Import button in the Forms Manager. A Select Import Source window appears, as shown in Figure 14-9. Click the Import ODBC Table Design button.

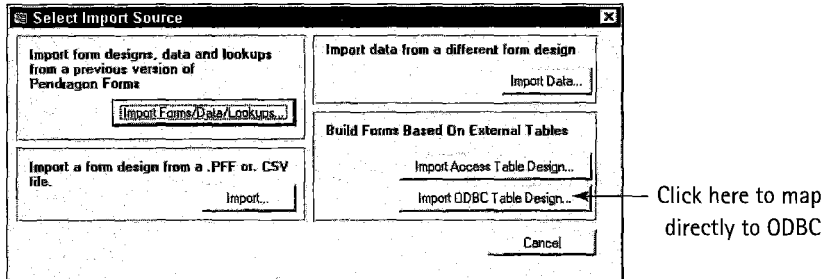


Figure 14-9: The Select Import Source window

A Select Data Source window appears. As shown in Figure 14-10, click the Machine Data Source tab, and then double-click the machine Data Source Name.

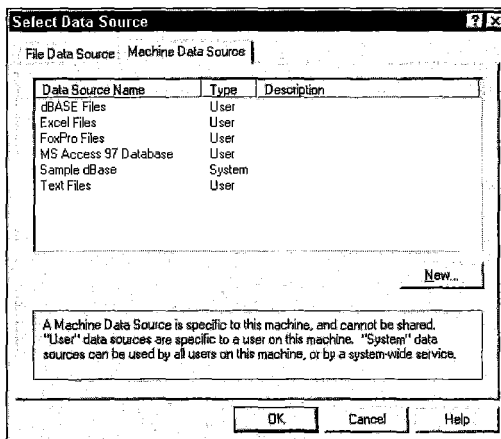


Figure 14-10: The Select Data Source window

If the ODBC database is password protected, you will be prompted to log onto the database.

A Table Selector window will display the list of tables in the database. Select a database table and click OK. You will then be prompted to enter a name for the Pendragon form that you are creating.

Editing the Form for Use on the Handheld

Once the Pendragon form has been created, either by your having created a linked table in the Forms3 database or by your having mapped directly to ODBC, you can edit the form design as necessary for use on the handheld.



Chapter 13, "Linking to an External Access Database," contains details on the type of edits that you may want to perform in order to make the form usable on the handheld.

Controlling Which Records Are Sent to the Handheld

The rules that determine which records from the external database table are sent to the handheld are set up in the Data Persistence section of the Form Properties window, and in the Additional Download Criteria field of the Advanced Form Properties window.



Refer to Chapter 13, "Linking to an External Access Database," for information on controlling records that are sent to the handheld. Also refer to Chapter 8, "Advanced Form Properties," for information on the Additional Download Criteria field.

Linking to the ODBC Database

After you have specified the criteria for sending records to the handheld, freeze the form design. In the Form Properties window, shown in Figure 14-11, click the Field Mappings button to access the Field Mapping Tool window.



The Field Mappings button is not active until you freeze the form design.

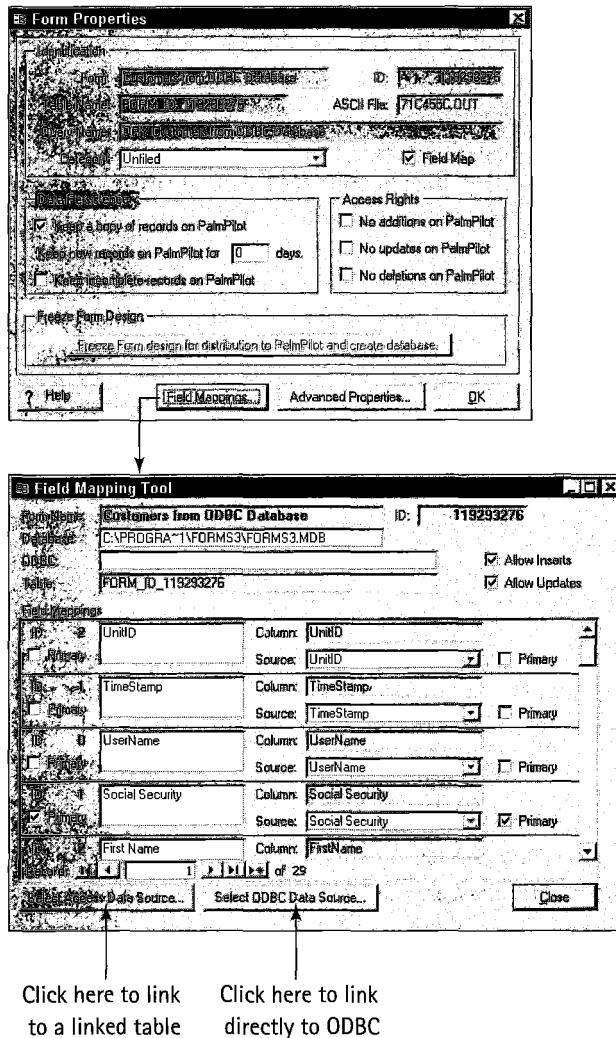


Figure 14-11: Click the Field Mappings button to view the Field Mapping Tool window.

If you have created a linked table in the Pendragon Forms database, click the Select Access Data Source button. Choose to link to the C:\Program Files\Forms3\Forms3.mdb file (or the Forms32k.mdb file if you are using Access 2000), and then select the linked table as the database table to which you want to link.

If you are mapping directly to ODBC, click the Select ODBC Data Source button. Select the machine Data Source Name, enter the database password if required, and select the database table to which you want to link.

In the Field Mapping Tool window, check that the column names of the form map to the correct Source column in the linked table. Also check that any field checked as primary on the form is also checked as primary in the source database table. If the form is missing a primary key, close the Field Mapping Tool window and edit the form by setting the Advanced Field Property of Primary in the appropriate field. Then redo the field mapping.

Once you are satisfied that the fields on the form have mapped to the correct database columns in the source database table, close the Field Mapping Tool window.

If you are mapping directly to ODBC, as you close the Field Mapping Tool window, you will be prompted: "Would you like to update the query to point to the new data source?" If you select Yes, you will be able to use the Edit/View button in the Pendragon Forms Manager to view data in the external database table. If you select No, then the Edit/View button will not work, and you will need to open the external database to view your data.

In the case of a linked table in the Pendragon Forms database, the Edit/View button in the Forms Manager does not display your data. Instead, you can view the linked table directly by bringing the Forms3: Database window to the foreground.



If you change any form properties or Advanced Form Properties, you will need to redo the field mapping to the external table.

Once you have linked the form to the external database table, you can distribute the form to the handheld.

Security Concerns When Using ODBC Databases

All handheld users need to have access to the Forms3.mdb database file – or Forms32k.mdb with Access 2000 – in order to synchronize.

Because handheld users have access to this file, there is therefore a risk that an individual can accidentally delete the Forms3.mdb (or Forms32k.mdb) file. It is therefore extremely important to back up the Forms3.mdb file (or Forms32k.mdb) on a regular basis.



For information on backing up the Pendragon Forms database, see Chapter 9, "Managing Data on the PC."

If you are using a linked table in Access to link to the ODBC database, your data will not be lost if the Forms3.mdb (or Forms32k.mdb) file is deleted accidentally. To enable handheld users to synchronize, you will need to restore a backup of the Forms3.mdb (or Forms32k.mdb) database.

Another security concern involves the use of passwords to access the back-end database – that is, your external ODBC database. If you are using linked tables in Access, the passwords to access the external ODBC database will be stored in the Forms3.mdb database (or Forms32k.mdb if you are using Access 2000). If you are mapping directly to ODBC, the password to access the external database will be visible in the Field Mapping Tool window in the Pendragon Forms Manager.

If handheld users' having access to passwords to the back-end database is unacceptable, you may want to consider using the WaveSync synchronization server as the synchronization mechanism. With WaveSync, the handheld user does not need to have access to the Forms3.mdb (or Forms32k.mdb) database in order to synchronize.



An evaluation version of WaveSync synchronization server is available on the CD-ROM.



For information about WaveSync, see Chapter 15, "Planning a Multi-User Installation." Installation instructions for the evaluation version of WaveSync can be found in Appendix C, "What's on the CD-ROM?"

Summary

This chapter outlined the two methods for linking a Pendragon form to an ODBC database table: creating a linked table in Access and mapping directly to ODBC. The security issues of which you should be aware when linking to ODBC were also outlined.

Part VI

Working in a Multi-User Environment

IN THIS PART

CHAPTER 15

Planning a Multi-User Installation

CHAPTER 16

Working with the Palm VII

Chapter 15

Planning a Multi-User Installation

IN THIS CHAPTER

- ◆ Configuring Pendragon Forms for multiple users
- ◆ Sharing records
- ◆ Understanding remote access issues
- ◆ Comparing synchronization options
- ◆ Addressing security concerns

ONCE YOU HAVE DESIGNED a form, you may want to deploy your form to everyone in your workgroup or organization in order to create a common database. This chapter looks at the issues that you need to consider in a multiuser deployment.

Configuring Pendragon Forms for Multiple Users

Pendragon Forms initially ships as a single-user license that allows one handheld device to synchronize with the database.

If you purchase a multiuser license for use with several handheld devices, you will need to enter a code to activate the multiuser license.

To activate a multiuser license:

1. Click the Options button in the Pendragon Forms Manager.
2. Enter your multiuser code.

Figure 15-1 shows the Options window where the multiuser license is entered.

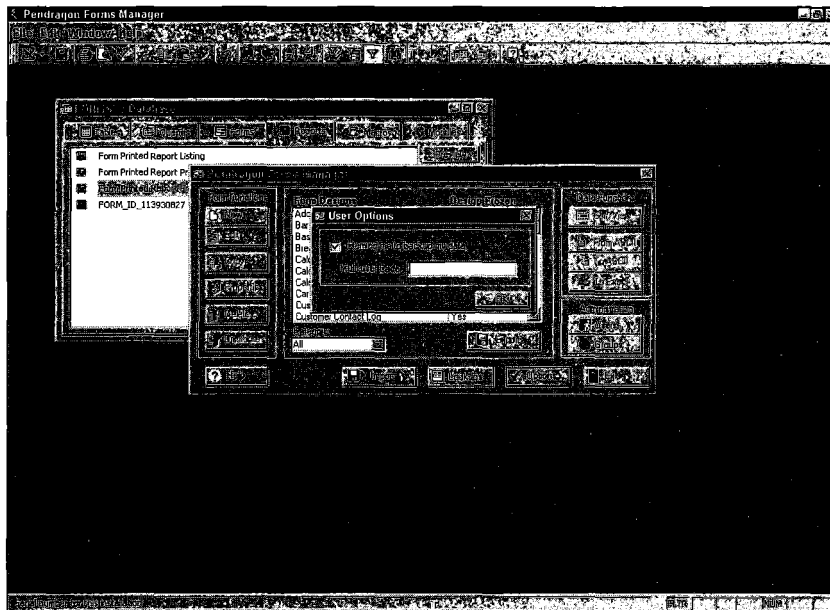


Figure 15-1: The Options window

After you have activated the multiuser license, you will need to click the Users button in the Pendragon Forms Manager to add the handheld user names to the list of active users. You will also need to click the Groups button to assign the handheld users to a User Group. Once you have assigned users to a User Group, any form that you assign to the group is received by each handheld during the next HotSync data transfer.



For information on the User List and User Groups, see Chapter 7, "Synchronization Rules."



Create a new User Group for handheld users, and use the Default User Group for administrators and individuals involved in designing forms. Members of the Default User Group receive every form that is distributed, whereas other handheld users will not need to receive every form.

Deciding Whether Records Are Shared

Assigning multiple handheld users to the same User Group allows the users to share the same form design.

Depending on the type of application that you have created, you may or may not want handheld users to share records as well.

In data collection applications such as taking surveys or performing inspections, each handheld user typically first goes into the field and creates new records and then performs a HotSync data transfer at the end of the day to send the data to the database. Although the users are using the same form design, they are not actually sharing the same records, because each user creates separate new records.

Similarly, in a work order scenario, where work orders are assigned to individual users, the users are sharing the same form design, but each person has his or her own records.

However, in the case of a group of physicians, for example, the physicians may prefer to share records, so that whoever is on duty has the list of all active patients.

By default, Pendragon Forms is configured so that each handheld user has separate records. The default Primary Key that is used to uniquely identify individual records includes a Username field that stores the handheld user name. In addition, the default criterion for downloading records from the PC to the handheld is that only those records that match the handheld user name are sent during a HotSync data transfer.

To allow handheld users to share records, you must first assure that users are able to receive all records, and that a record created by one user is able to be updated by another user.

There are two things that you need to do to allow sharing of records:

1. Choose a Primary Key for the form that does not include the Username. If you do not choose your own Primary Key, a duplicate record will be created in the database when one user modifies a record that was created by another user.
2. Delete the default Additional Download Criteria in the Advanced Form Properties window that sends only records that match the Username. This lets all users receive all records.



For information on Primary Keys, see Chapter 6, "Advanced Field Properties."
For information on Additional Download Criteria, refer to Chapter 8, "Advanced Form Properties."



If you allow multiple users to share records, there is a risk that if two users modify the same record and then both perform a HotSync data transfer, the data from the first handheld to synchronize will be overwritten by the second. This risk can be minimized if the users adhere to a synchronization schedule. For example, a physician working a day shift has to synchronize at the start of his or her work day to receive record updates from the evening shift, and then has to synchronize a second time at the end of the day shift to update the database for the next physician coming on duty.

Records should be allowed to be shared only if the possibility that one user may overwrite the other is low enough to be tolerated. If you do not want to tolerate that risk, you should not allow users to share records.

One approach to the example of physicians' sharing a patient list is that the patient information can be stored in a parent form, and then each patient visit can be in a subform. Each physician can view the subform to see a history of the patient visits but does not modify the historical records. Instead, a physician who sees a patient on her shift can create a new subform record for that patient. That way, if two physicians see the same patient on the same day, there will be two subform records in the database.

Enabling Remote Access

Remote access is the ability to perform a HotSync data transfer without the handheld's being in physical contact with the PC.

Remote access can be achieved by connecting a modem to the Palm device and dialing in to the PC to synchronize. The PC that the Palm device is dialing needs a modem and a dedicated phone line, and the HotSync Manager software has to be running. Alternatively, you can accomplish remote access by dialing up to a network and synchronizing to a PC on the network.

In a multiuser environment, you need to consider the location of the handheld users, and whether they have access to a PC that can run Pendragon Forms. For instance, if your handheld users perform fieldwork and are unable in the course of a day to return to the office to synchronize, you may want to consider enabling remote access.

The HotSync Manager software supports only one user synchronizing at a time. Therefore, if you want to enable remote access, you will need to consider the number of handheld users that you have, the times of day that the users are likely to synchronize, and the length of time that each synchronization takes. If users often receive a busy signal when they attempt to synchronize remotely, they are not likely to want to continue using the handheld.

Remote access is an issue to keep in mind when considering your synchronization options.

Choosing the Right Synchronization Option

One of the biggest issues in a multiuser deployment of Pendragon Forms is the question of how the handheld users will synchronize their data with the PC.

In some cases it is acceptable to use the HotSync Manager software, for instance if it is unlikely that two people will need to synchronize at exactly the same time. In other cases you may prefer to look at alternative products that allow multiple simultaneous HotSync data transfers.

There are several synchronization options to consider:

- ◆ Synchronizing to one PC
- ◆ Synchronizing over a LAN using the Pendragon Forms Network Setup (Nsetup)
- ◆ Synchronizing over a LAN to a server running 3Com's Network HotSync
- ◆ Using the WaveSync synchronization server from WaveWare Communications (an evaluation version is included on this book's CD-ROM)

Each of these options is discussed in turn.

Synchronizing to One PC

Synchronizing to one PC works best if you have a small user base and all the users are able to return to the "base station" at least once a day to perform a HotSync data transfer.

In this case, Pendragon Forms is installed on one PC and all handheld users perform a local HotSync data transfer by placing their handheld devices in the HotSync cradle one at a time. You can allow synchronization via modem if the users do not all need to synchronize at the same time of day.

Synchronizing to one PC works well if individuals work different shifts or return to the office at slightly different times, so that there is never a queue to perform a HotSync data transfer.

Synchronizing Over a LAN Using Pendragon Forms Network Setup

If users have PCs that are connected via a local area network (LAN), it may be more convenient for the users to perform a HotSync data transfer at their desks than to synchronize at one dedicated station.

When you purchase a multiuser license for Pendragon Forms, you can obtain a Network Setup utility that allows you to install Pendragon Forms on a server, and

then run the Network Setup from each workstation PC to install the Pendragon Forms conduit. The HotSync Manager software has to be installed on each workstation.

With the Pendragon Forms Network Setup, multiple handheld devices can synchronize simultaneously at different workstations on the LAN. Each HotSync data transfer sends data from the handheld directly to the Pendragon Forms database on the server.

If the Pendragon Forms database is being stored on a Windows NT server, remote access is generally not possible by directly dialing in to the server. This is because a user must be logged into the Windows NT server before the HotSync Manager can be switched on, and most Windows NT servers are configured to run without any users being logged in (for security reasons). Instead, if the handheld user can dial into a workstation running Windows 95 or Windows 98, a Modem HotSync can take place.

Figure 15-2 illustrates an example of how a network can be set up to use Pendragon Forms Network Setup.

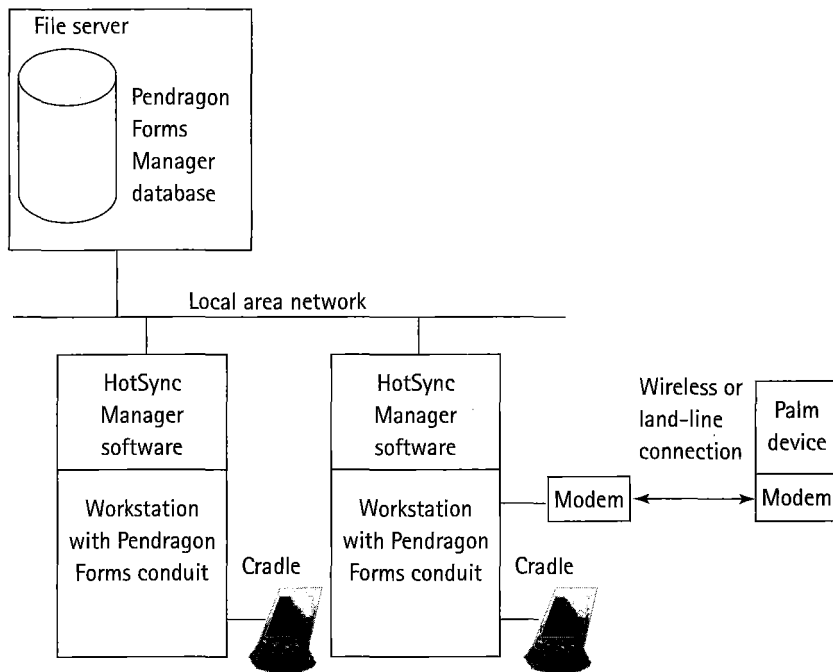


Figure 15-2: Network Configuration for Pendragon Forms Network Setup

Synchronizing Over a LAN Using 3Com's Network HotSync

Another approach to using Pendragon Forms on a LAN is to use 3Com's Network HotSync product. Pendragon Forms is installed on a server, and Network HotSync is installed on each local PC in addition to the HotSync Manager. The LAN must support the TCP/IP protocol to use Network HotSync.

With Network HotSync, although two handheld devices can appear to synchronize at the same time, the product actually allows one handheld to synchronize and puts the other handheld in a queue to synchronize next.

Remote access does not require the handheld user to dial in to the workstation directly. If the corporate LAN has a Remote Access Server (RAS server), the handheld user can dial into the RAS server and can then perform a Network HotSync to the server where Pendragon Forms is stored.

However, like the standard HotSync Manager software, Network HotSync requires a user to be logged in and typically would not be run under Windows NT.

The main reason to use Network HotSync instead of the standard HotSync software is to reduce the amount of synchronization data flowing over the network. When the Pendragon Forms conduit is synchronizing with the Pendragon Forms database, large chunks of the database may need to be read into memory. For example, the Pendragon Forms conduit must identify which records need to be sent to the handheld. If the Pendragon Forms database is large, several megabytes of data may need to be read into the memory of the PC on which the conduit is running, even when a very small amount of the data is actually transferred to the handheld. This is not a problem when the conduit is running on the machine where the database is stored, because hard drives are very fast, but if the database is on a network, there may be a noticeable delay while data is read across the network.

The typical small LAN uses links that transfer data at 1 or 2 megabytes per second. This data transmission rate is shared by all the users on the network, so the more users who access the network at the same time, the slower the link will seem to each user. Also, some parts of the network may have slower links, such as ISDN (0.02 megabytes per second) or T1 (0.15 megabytes per second). With slower network links, or with overcrowded networks, Network HotSync can help. With Network HotSync, synchronization takes place on the same server as the database. After sorting through the database records on the server, Network HotSync relays only the requested records to the workstation.

Figure 15-3 shows an example of a network configured to use Pendragon Forms with 3Com's Network HotSync.

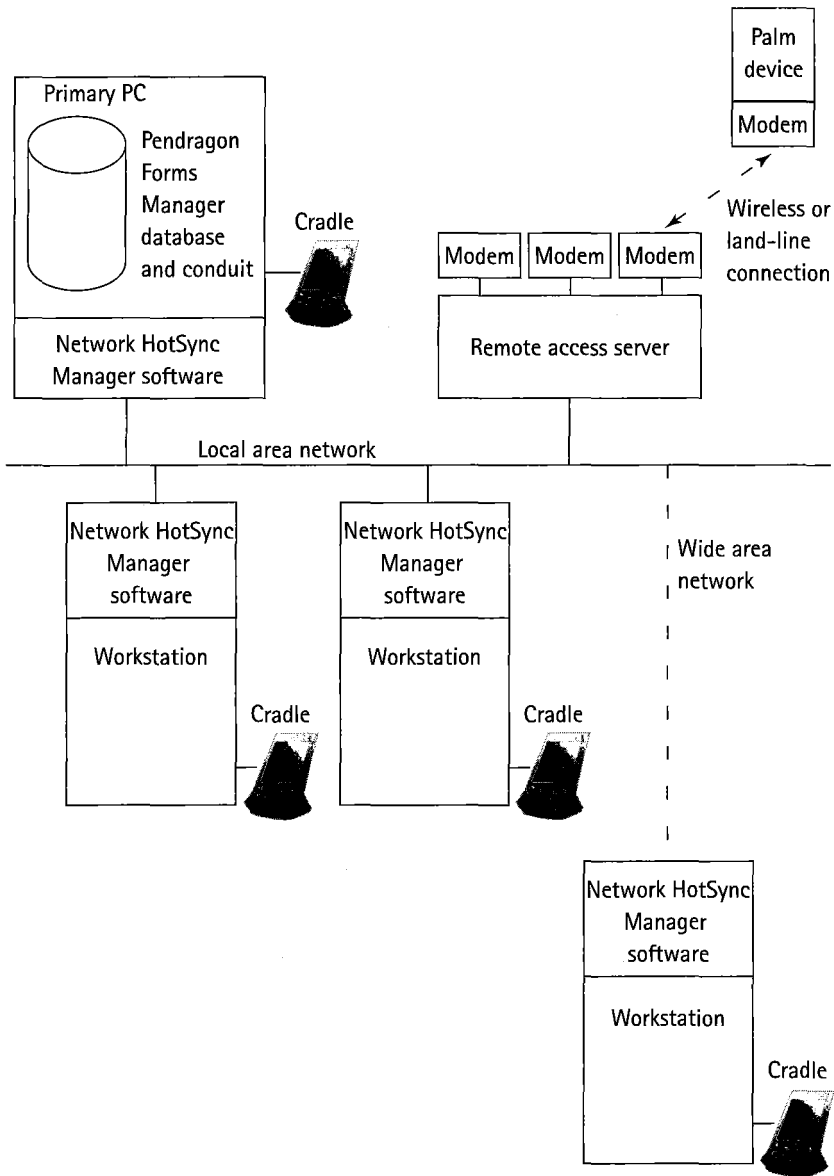


Figure 15-3: Network Configuration for Pendragon Forms and 3Com's Network HotSync

Using the WaveSync Synchronization Server from WaveWare Communications

The WaveSync synchronization server is a replacement for the HotSync Manager that allows multiple users to synchronize simultaneously.

In this case, Pendragon Forms and the WaveSync software are installed on a Windows NT server. WaveSync proxy software is installed on the workstation PCs instead of the HotSync Manager. As with Network HotSync, communications with the handheld are relayed by the proxy software to the server, and your network must support the TCP/IP protocol.

The WaveSync server software is specifically designed for handling multiple users on a corporate network and has many advantages over the HotSync Manager and Network HotSync Manager software:

- ◆ Multiple handheld devices can synchronize simultaneously.
- ◆ The WaveSync server uses Windows NT security to authenticate users before they synchronize.
- ◆ Conduits can be centrally managed, and handheld applications can be installed and uninstalled under central control.
- ◆ Synchronization can be up to five times faster than with Network HotSync when synchronizing via modem.

If you are implementing remote access for more than a few users, a WaveSync server may be your only solution to avoid busy signals or long HotSync sessions. Also, if your organization uses slower network links (such as for branch offices), a WaveSync server will significantly reduce synchronization time.

Information about the WaveSync server can be found at www.waveware.net.



An evaluation version of the WaveSync synchronization server is available on the CD-ROM.

Figure 15-4 shows an example of a network configured to use Pendragon Forms and WaveSync synchronization server.

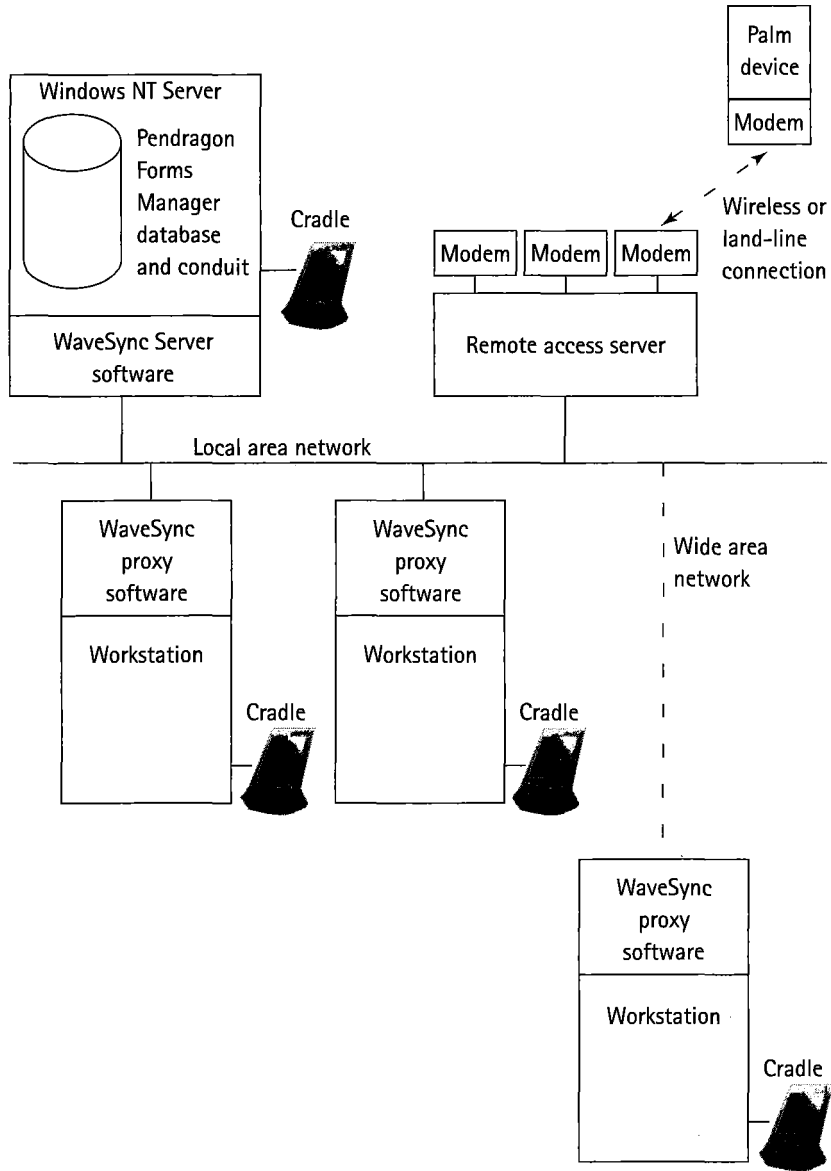


Figure 15-4: Network Configuration for Pendragon Forms and WaveSync synchronization server

Security Issues to Consider

In many cases, administrators may be comfortable with little or no security, especially with small, close-knit user groups. For example, a small office of professionals who share data on a peer-to-peer network already accept the risks associated with sharing their files and trust their coworkers implicitly.

However, in an enterprise where the data is proprietary or extremely valuable, it makes sense to impose security measures.

There are several security issues to consider when creating your data collection application.

- ◆ **Access to sensitive data:** You may want to limit access to the central database by users with handhelds, or users on the network.
- ◆ **Deletion/corruption of data:** Though there may be a minimal risk of malicious damage, it may be possible for users to delete or corrupt data on the server.
- ◆ **Deletion of database files:** Microsoft Access is a file-based database. This means that all users who access it must have read/write access to a .mdb file, and this in turn means that users may be able to delete the entire database file via file management tools.
- ◆ **Access to external databases via the Pendragon Forms database:** In order to synchronize with password-protected external database systems, the Pendragon Forms database generally needs to include the database user's name and password in the Forms3.mdb file (or Forms32k.mdb if you are using Microsoft Access 2000). This is true whether using the Access linking feature or using a direct ODBC field map. Unfortunately, this means that users who gain access to the Forms3.mdb (or Forms32k.mdb) file will generally gain some access to the external databases too.
- ◆ **Impersonation ("spoofing"):** It is relatively easy to change the name assigned to a Palm device to impersonate another user. By performing a hard reset and performing a synchronization, users can rename their Palm devices quite easily. If your Pendragon Forms applications send data from the desktop to the Palm (such as sales reports), you may need to defend against this type of attack. Because the Palm user name is the mechanism used to control which users see which datasets, spoofing may allow a user to see data that was only meant for another user.

Taking Precautionary Measures

You can take several steps to improve the security of the system.

- ◆ Password protecting (and optionally encrypting) the Forms3.mdb (or Forms32k.mdb) file will prevent unauthorized users from easily reading the contents of the database via the network. The Microsoft Access security system can also prevent users from accessing, writing to, or deleting certain tables. It also protects against spoofing. However, it does not protect the Forms3.mdb (or Forms32k.mdb) file itself from being deleted.

When you set a password on the Forms3.mdb file, the Access user name and password must be set on the handheld via the Login screen. To access the Login screen, tap the Forms icon, and then tap the handheld Menu button and select the menu option Login (or use the Graffiti shortcut /O) to enter the Access user name and password. Using a password on the database means the hacker using a spoofing attack cannot simply rename his Palm device to gain access to another user's data, because the hacker does not know the database user name and password.

- ◆ Storing your data on a server database provides better protection against database deletion than storing your data directly in the Pendragon Forms database. Because only the database system administrator can delete a server database, end users cannot delete the entire database. Using a server database is accomplished by upsizing the data tables to a server database and replacing the original tables in the Forms3.mdb (or Forms32k.mdb) file with links to the corresponding server database tables. In fact, Microsoft provides an upsizing utility that will convert Access database tables to Microsoft SQL Server tables automatically.

Upsizing the data tables still leaves the data itself vulnerable to users who gain access to the Forms3.mdb (or Forms32k.mdb if you are using Access 2000) file. Also, the .mdb file may still be deleted. Deletion of the .mdb file will not harm the collected data, because it is stored on another server, but configuration information such as form designs and user lists will be lost.

- ◆ Completely replacing the Forms3.mdb (or Forms32k.mdb) file with a server database eliminates the risk of deletion of the .mdb file. In this case all of the tables in the Forms3.mdb (or Forms32k.mdb) file are upsized to a server database, including the Pendragon Forms configuration tables (form, question, user_group, and so on).

Because all server databases will require a user name and password to log in, the password to the server database needs to be entered on the handheld to allow the conduit to access the database server. This also protects the database from spoofing attacks.

- ◆ Using a WaveSync synchronization server provides all-around improved security. The WaveSync server runs on a Windows NT server and replaces the functionality of the HotSync Manager software that would normally be installed on users' desktops.

It is possible to set up the WaveSync server so that handheld users cannot see the database directly via the file system. This protects the database against file deletion and hides passwords to external databases behind the administrator's Windows NT security.

Security Limitations

Data on the handheld is not encrypted. This means that if the Palm hardware is compromised, so is the data it holds.

A number of third-party products allow the Palm device to automatically become password protected when powered off. These tools make it more difficult, though not impossible, for hackers to access the memory of the device.

Another security limitation is that Network traffic is not encrypted. "Packet Sniffers" can be used to intercept packets of information sent to and from stations where users are synchronizing. This is generally not an issue for users synchronizing on a local area network (LAN) but may be an issue for users synchronizing via the Internet.

Summary

This chapter outlined the issues that you need to consider when planning a multiuser installation of Pendragon Forms. The Pendragon Forms Manager has to be configured for multiple users, and you need to select the synchronization option that best suits the needs of your handheld users. Issues such as remote access, security, and performance all factor into the way that you implement a multiuser solution.

Chapter 16

Working with the Palm VII

IN THIS CHAPTER

- ◆ How the Palm VII works
- ◆ Choosing Pendragon Forms or Palm Query applications
- ◆ Creating a Transmit button
- ◆ Designing a Palm VII-compatible Web site

LAUNCHED IN THE SECOND quarter of 1999, the Palm VII is one of the newest additions to the family of Palm computers. Slightly bigger than a Palm III, the Palm VII includes a radio modem that gives you a wireless connection to the Internet.

How the Palm VII Works

The estimated battery life of a Palm VII is one to two weeks, using a pair of AAA batteries. While most other Palm Computing organizers may run two to four times longer on the same set of batteries, the Palm VII has excellent battery life when compared with other two-way, wireless devices such as cellular phones. To accomplish this comparatively long battery life, the designers of the Palm VII have limited its ability to remain connected to the Internet for more than a few seconds at a time. Instead of being able to “surf” the Web, sending and receiving a constant stream of packets of information, the Palm VII is designed for intermittent Web connection. Typically, you flip up the antenna of the Palm VII, it transmits a packet of information from the handheld application that you are running, and then it waits to receive a packet in response. The transmitter on the Palm VII may be switched off while the receiver is waiting to receive a packet of information. The intermittent nature of transmitting packets saves on battery life.

The built-in applications that ship with the Palm VII are designed to look up small items of information, such as the weather forecast for a given city, or movie times in a particular ZIP code. 3Com refers to this as Web clipping, not Web browsing.

Palm VII applications are like static Web pages that are preloaded onto the handheld. Transmitting a request from one of these applications is similar to clicking a hyperlink or submitting a form in your Web browser to link you to a live Web page. The Palm VII displays the page returned by the Web server.

Can I Synchronize Wirelessly with a Palm VII Instead of Performing a HotSync?

The short answer is no. The wireless capability of the Palm VII is not a substitute for a direct (local) HotSync data transfer or a modem HotSync.

The only protocol that the Palm VII can use wirelessly is an HTTP protocol for requesting Web pages. There is no mechanism for performing a HotSync data transfer. Even if such a mechanism existed, the Palm VII would not have the battery power to sustain a wireless HotSync data transfer. Finally, because there is a monthly fee for wireless access via the Palm VII, it would be expensive to synchronize wirelessly on a regular basis.

Choosing Pendragon Forms or PQAs

The Palm VII ships with software that enables you to create your own Palm Query Applications or PQAs. This means that you can create HTML files that can be loaded onto the Palm VII and can be used to send information to your Web site.

If you want to create an application to request current information based on one or two fields of user input, then a PQA may be the best solution for you. However, there are several reasons why you might want to use Pendragon Forms with the Palm VII instead of using the standard PQA Builder software that ships with the Palm VII hardware.

Web forms that you can create in HTML do not have built-in field validation. Instead, the Web server that responds to your request can perform validation. This means that if you create a form on a Web page, and you want the user to enter a number in a specific range, the Web page has no way to verify the validity of the number that was entered before the request is sent. A Pendragon form, on the other hand, can do sophisticated validation before the request is sent. The ability to validate a page before making Web requests can save both money and battery life on a Palm VII.

Entering data in a Pendragon Form also allows the handheld user to keep a history of the records entered. In contrast, entering data into a Web page in a PQA does not keep a copy of the data entered, although a limited history of Web server responses is maintained (similar to your PC's keeping a history of Web pages that you have visited, but periodically discarding from the list). Pendragon Forms is therefore useful if your handheld users need to maintain records on the handheld for a period of time for reference purposes.

Palm VII Alternatives

There are several alternative ways to wirelessly connect Pendragon Forms to a server. The alternatives operate by using a form of network synchronization (either Network HotSync, or, for best performance and reliability, WaveSync server software).

Novatel Wireless manufactures the Minstrel III wireless modem specifically for Palm Computing devices. The Minstrel III has battery life comparable with that of a cell phone, so it may require daily recharging.

It is also possible to purchase adapters that enable a Palm organizer to connect wirelessly with a cell phone. These adapters may be convenient if you already carry a compatible GSM phone.

Finally, Qualcomm has a cellular phone with a built-in Palm III organizer, called the pdQ smartphone. The built-in Palm OS device will be able to synchronize via the radio modem of the phone.

These alternatives enable full synchronization but are physically larger than a Palm VII and may require daily recharging.

If you are in a region where the wireless service for the Palm VII is not immediately available, you might want to enter data into a Pendragon Form and then transmit the record at a later time. A standard PQA application may not give you the ability to store one or more records until you are able to transmit the data.

Pendragon Forms is also useful if you have data in a database that you want to download to the handheld via HotSync, then using the Palm VII to perform live queries on a selected record while in the field. For example, you may download a large list of packages when you synchronize at your desktop, but have on-demand, real-time delivery tracking information made accessible via the Palm VII wireless capability. To make this type of application work, you would need to have a Web database on your Web server.

Creating a Transmit Button

Pendragon Forms uses a `transmit palmnet` scripting command to send a record to a Web site via the Palm VII.

When using the Palm VII, it is best to give the handheld user control of when a record is transmitted. Sending a wireless transmission can take several seconds (perhaps 15 seconds), and if the transmission takes place without the handheld user's knowledge, the user might interpret the lack of immediate response in the

program on the handheld as a hardware problem. Instead, it is better for the handheld user to initiate the transmission.

To give the handheld user control of transmitting a record via the Palm VII, the `transmit palmnet` scripting command should be used only in a `click` event script in a Button field. The handheld user can then have time to raise the Palm VII antenna before clicking the button to transmit the record.

Figure 16-1 shows a form with a Transmit button. To create the label on the button field, type the word `Transmit` as the Default Value in the Advanced Field Properties window of the Button field.



For information on the Default Value, see Chapter 6, "Advanced Field Properties." For general information on scripts, see Chapter 11, "Using Scripts."

Figure 16-1: A form with a Transmit button

The format of the `transmit palmnet` scripting statement in a Click event is:

```
click:
transmit palmnet "address"
```

The word *address* refers to a Web address. Depending on how you define the address parameter, it is possible to send the contents of the current record to a Web site. The Web server software can be written to respond with a Web page that contains the results of a database search, or the Web server software can store the record received and respond with a Web page that confirms receipt of the record.

If the address parameter is a simple address, such as

```
click:
transmit palmnet "http://www.ourcompany.com/saleprices.htm"
```

the Palm VII will simply display the contents of the Web page at `http://www.ourcompany.com/saleprices.htm`. Note that in this case no information is being sent from the Pendragon form to the Web site. In this example, the script in the Button field enables the handheld user to receive a live Web page update of the day's sale prices that the handheld user can then use to inform a customer.

If the address parameter in the `transmit palmnet` statement contains a question mark at the end of a CGI program name and ends with an ampersand (&), Pendragon Forms will append the data in the current record to the Web address URL before transmitting. The data is written using the Internet standard "Quoted Printable" format that is used for encoding data in URLs.

Data from the Pendragon Forms record is appended to the Web address in the following format:

```
FID=<formID>&TS=<timestamp>&Q1=<field value>&Q2=<field
value>&Q3=<field value> . . .
```

- ◆ FID stands for Form ID; it is followed by the Form ID number of the current form.
- ◆ TS stands for TimeStamp; it is the creation date of the record expressed as the number of seconds since 01/01/1904.
- ◆ Q1 means Field 1 on the form, followed by the value in that field, Q2 means Field 2, and so on.

For example, if the form in Figure 16-1 contained the following script in the Button field:

```
click:
transmit palmnet "http://www.ourcompany.com/cgi-
bin/getinfo.cgi?check=1&"
```

then the actual URL that would be sent to the Web server would look like:

```
http://www.ourcompany.com/cgi-
bin/getinfo.cgi?checkqty=1&FID=23456&TS=32416345&Q1=8%2F30%2F99+2%3A
54+PM&Q2=Malinde&Q3=Starry+swimsuit&Q4=12&Q5=Transmit
```

Once the data has been sent to the Web server, the response that the server sends back to the Palm VII depends on the way that the `getinfo.cgi` script is written. For example, the `getinfo.cgi` program can return a confirmation number that the record has been received, or it can display a status page to notify if an item is in stock.

Figure 16-2 shows the type of response that can be sent from the Web server to the Palm VII. When the handheld user taps the Back button, he or she will return to the Forms application on the handheld.

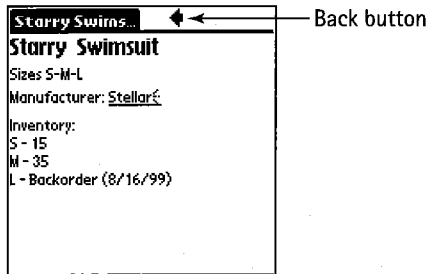


Figure 16-2: The Palm VII displays a response from the Web server.

Another way to specify the address parameter in a transmit palmnet scripting statement is to substitute arguments in the Web URL for values in various fields on the form. The value in a field is referenced by the field number placed between two dollar signs. For example, \$17\$ means Field 17.

Figure 16-3 shows a form that is used to check if a customer's account is in good standing for a particular credit amount. Only the customer account number and the order amount need to be submitted for the credit check to occur.

Customer Name:	P. McOtter
Account#:	481240
Date:	8/29/99
Order Amount:	\$24,000.00
Credit Check:	(Check)
Approved?	<input type="checkbox"/>

End ⏪ ⏩ ⏴ ⏵

Figure 16-3: A form for checking credit status

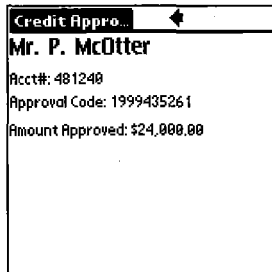
In order to send two fields (Customer Account and Order Total) to the Web server, the script in the Button field is:

```
click:
transmit palmnet "http://www.ourcompany.com/cgi-
bin/creditcheck.cgi?acct=$2$&amount=$4$"
```

The actual Web URL that is transmitted looks like:

```
http://www.ourcompany.com/cgi-
bin/creditcheck.cgi?acct=481240&amount=%2424%2C000%2E00
```

Figure 16-4 shows the type of response that the Web server can send to the Palm VII. To return to the Forms application, the handheld user has to tap the Back button on the Web page response.

A screenshot of a handheld device screen showing a credit approval response. The screen has a title bar that says "Credit Appro..." with a back arrow icon to its right. Below the title bar, the text reads: "Mr. P. McOtter", "Acct#: 481240", "Approval Code: 1999435261", and "Amount Approved: \$24,000.00".

Credit Appro...
Mr. P. McOtter
Acct#: 481240
Approval Code: 1999435261
Amount Approved: \$24,000.00

Figure 16-4: A Web server response to a credit-checking query

At present, Pendragon Forms does not have direct access to the response that the Web server sends to the Palm VII. This means that, for example, the script cannot determine if the Web request was sent successfully or extract data from the server's response. You can add a field to your form that the handheld user can fill in manually after receiving the Web server response. In Figure 16-3, a Yes/No Checkbox field enables the handheld user to mark whether an approval was received from the Web server.

How Much Data Can I Transmit with a Palm VII?

When the Palm VII was first released, basic pricing for the wireless Palm Net service was about \$9.99 per month for 50KB of combined transmitted and received data. This corresponds to approximately five requests per day. The expanded plan pricing was \$24.99 per month for 150KB or approximately 15 requests per day, assuming that pages are optimized for the Palm VII screen.

A typical PQA application only contains one to five fields — for example, a stock quote request requires only a ticker symbol to be entered. A monthly limit of 150KB is a lot if you are requesting only one stock quote per day.

In comparison, a Pendragon form can contain many more than five fields. The size of a record transmitted from Pendragon Forms is proportional to the number of characters in the Web URL that is sent using the Transmit Palmnet scripting statement. If you wanted to transmit a form with 50 fields, one record would be ten times larger than the average PQA request. In considering the Palm VII as a solution with Pendragon Forms, therefore, you will need to look at the cost of the Palm Net service and the battery life of the Palm VII before you deploy to multiple handheld devices.

Designing a Palm VII-Compatible Web Site

When a Palm VII transmits a request, the default size of the response contains the first 500 bytes of information from the requested Web page. This small size is optimized for use on the Palm VII, although you can add tags within your HTML to enable the Palm VII to receive larger Web pages.

3Com provides developer guidelines on creating Web content for use with the Palm VII. These guidelines can be found in the developer link at www.palm.net.

If you intend to use Pendragon Forms with the Palm VII, you will probably need to create a custom Web site that can accept the data transmitted from Pendragon Forms. CGI programs for Web servers are typically written using programming tools such as Perl, C, Microsoft Active Server Pages, and Allaire Cold Fusion. Web site design is beyond the scope of this book, but you can refer to many other books on the topic of CGI programming.

Summary

This chapter describes the main features of the Palm VII and its innovative wireless protocol.

Palm Query Applications can provide quick and easy solutions for applications that submit only a few fields. Pendragon Forms can provide improved validation and connect offline data with real-time Web information.

By using button fields with scripts you can access Web pages and Web databases by transmitting requests that include data from fields on a form.

Appendix A

Troubleshooting Tips

Installation Troubleshooting

Refer to the following table if you have problems installing Pendragon Forms.

Problem

During installation, the Setup program freezes and installation is never completed.

When I try to open the Pendragon Forms Manager program, I get a runtime error message saying "Cannot find project or library."

Solution

Check if you are using antivirus software. Some older versions of antivirus software are not able to handle scanning large databases.

An interim solution is to disable the automatic virus-checking of Access databases. You can still manually virus-check Access databases.

A permanent solution is to refer to the vendor of your antivirus software to see if they have an update that will fix the problem.

If you have the full version of Microsoft Access, a component of Access called Data Access Objects may not have been installed when Access was installed on your PC.

Close the Pendragon Forms Manager, then rerun the Microsoft Office Professional CD-ROM and select to Add/Remove components for Microsoft Access. Check the box to add the component Visual Basic Data Access Objects. If this box is already checked, follow the instructions to register the Data Access Objects component in the next paragraph.

Continued

Continued

Problem

When I click a button in the Forms Manager, I get an error message: "The expression ON CLICK you entered as the event property setting produced the following error: Automation error."

Solution

If you do not have the full version of Microsoft Access, or if you have Access and Data Access Objects already installed, the problem may be that the Data Access Objects component is not registered. Close the Pendragon Forms Manager. Click Start → Programs → Pendragon Forms 3.0 → Configuration Tool. In the Configuration Tool window, click the Diagnostics tab and click the button labeled Re-register Data Access Objects. Figure A-1 shows the Configuration Tool window.

This error typically occurs when a shared component called comdlg32.ocx is not registered.

To register this component, click Start → Programs → Pendragon Forms 3.0 → Configuration Tool. In the Configuration Tool window, click the Diagnostics tab and then click the button labeled Re-register Common Dialogs. Figure A-1 shows the Configuration Tool window.

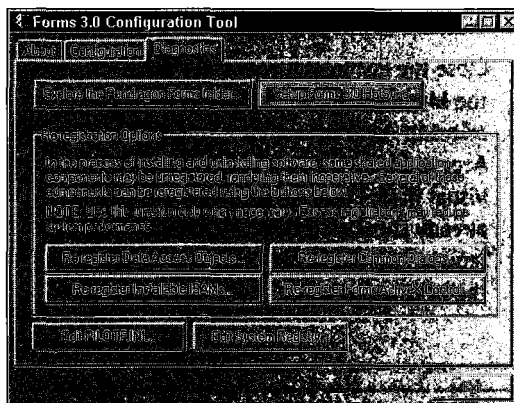


Figure A-1: The Pendragon Forms Configuration Tool window

Sending Forms to the Palm Device

If you distribute a form to the handheld and then perform a HotSync data transfer, but the form does not appear on the handheld, check the sections that follow.

Is There a Forms Icon on the Handheld?

If you tap the Applications button on the handheld and select to display All applications, there should be a Forms icon on the Palm screen, as shown in Figure A-2.

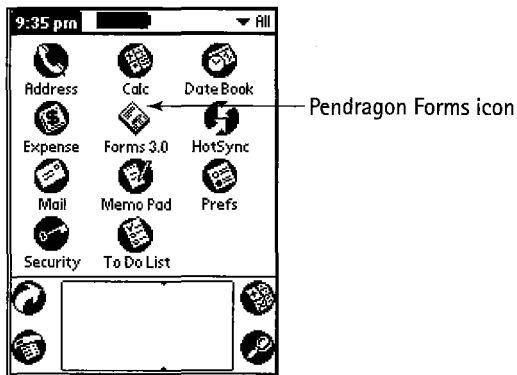


Figure A-2: The Forms icon must be on your Palm before you can send form designs to the handheld.

If you do not see a Forms icon, then on your PC click Start → Programs → Pendragon Forms 3.0 → Install Forms 3.0 on Handheld.

The Palm Install Tool program will run.

- ◆ Verify that the program selected to be installed is Forms3.prc. This file is typically in the C:\Program Files\FORMS3 folder.
- ◆ Also verify that your correct handheld user name has been selected as the device to which the Forms program will be installed. To check the user name of a handheld device, tap the HotSync icon on the Palm device, and the user name will appear in the upper-right corner of the HotSync screen, as shown in Figure A-3.

Once you are satisfied that the Forms3.prc program has been selected for the correct handheld user, click the Done button. Perform a HotSync data transfer to install Forms3.prc on your handheld.

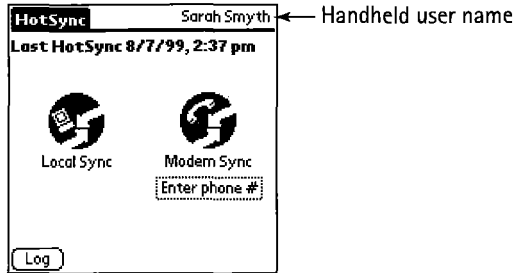


Figure A-3: To check the handheld user name, tap the HotSync icon on the Palm device.

Once the Forms icon is installed, you can distribute form designs from the Pendragon Forms Manager and perform another HotSync to send the form designs to the handheld.

Is the Pendragon Forms Conduit Registered with the HotSync Manager?

If the Forms icon is on your Palm organizer and you still cannot send form designs to the handheld, check if the Pendragon Forms conduit is registered. Third-party applications such as Pendragon Forms must be registered with the HotSync Manager in order to synchronize during the HotSync data transfer.

To check if Pendragon Forms is registered, right-click the HotSync Manager icon in your Windows system tray. Choose Custom from the menu that appears.

Figure A-4 shows the HotSync Manager Custom window. Select your handheld user name, and then check if the Pendragon Forms 3.0 conduit is in the list of applications to synchronize. The action for Pendragon Forms should be Synchronize the Files.

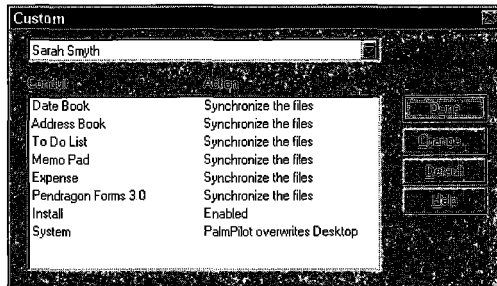


Figure A-4: The HotSync Manager Custom window

If Pendragon Forms is not in the list of conduits to synchronize, do the following to reregister the Pendragon Forms conduit:

1. Close the HotSync Manager Custom window.
2. Right-click the HotSync Manager icon and choose Exit.
3. Click Start → Programs → Pendragon Forms 3.0 → Configuration Tool.
4. Click the Diagnostics tab, and then click the Setup Forms 3.0 HotSync button.
5. Close the Configuration Tool window. Restart the HotSync Manager by clicking Start → Programs → Palm Desktop → HotSync Manager. The HotSync Manager icon should reappear in your Windows system tray.

You can then open the Pendragon Forms Manager, distribute a form, and perform a HotSync data transfer. During the HotSync process, you should see the message “Synchronizing Pendragon Forms.” This indicates that the Pendragon Forms conduit is working.



If you reinstall the Palm Desktop software, the new HotSync Manager that you install will not be aware of the Pendragon Forms conduit. You will need to reregister the Forms conduit as described previously.

Did the HotSync Process Generate an Error Message in the HotSync Log?

Often, if your form cannot be distributed to the handheld, the Pendragon Forms conduit will generate an error that appears in the HotSync Log.



See the topic “HotSync Log Error Messages” in this appendix.

Freezing a Form

If you attempt to freeze a form design and the Pendragon Forms Manager does not allow you to do so, there may be a problem with your form design. Common

error messages that occur when freezing a form are shown in the following table. Once you have made the appropriate changes to your form design, retry freezing the form.

Error Message Generated When Attempting to Freeze Form Design

Solution

"A form must have at least one field."

If you create a form that has a name but no fields, the form design cannot be frozen.

Edit the form and add at least one field to the form before attempting to freeze the form.

"Duplicate database column names are not allowed."

When a form is frozen, a database table is created for storing the data associated with the form. For each field on your form there is a column in the database table for storing your records. Microsoft Access does not allow two database columns to have the same name.

Edit the form and check the Advanced Field Properties for each field. If two fields have the same Column Name, change one of the names to be different.

"Missing field name."

A field is missing a field name but has Popup options, or a script.

Edit the form and add a field name to the field.

"Every section requires a name."

If you have a section field on your form, the field must have a field name as well as a section name.

The error message specifies the field that is missing a Section name. Close the Form Properties window, click the name of the form, and then click the Edit button to open the Form Designer window. Display the field in the Form Designer window, and then type a Section name in the Section field. Save the changes to the form and reattempt freezing the form.

"Lookup List not specified."

If you have a Lookup List or Exclusive Lookup List on your form, you must select a Lookup List that will be displayed when the handheld user is in the Lookup List field.

Error Message Generated When Attempting to Freeze Form Design**Solution**

	<p>The error message includes the name of the field that has the problem.</p> <p>Close the Form Properties window and edit the form design. In the Lookup List field, select the Lookup List that you want to use. If you have not yet created a Lookup List, you will need to create one before you can reference it in your form. Once you have referenced a Lookup List in the field, you can reattempt freezing the form design.</p>
"Popup List (or forms/sections list) is empty."	<p>The field specified in the error message is a Popup List, Jump Popup, Subform list, or Multi-Selection list that contains no options.</p> <p>Edit the form and add the options that you want to appear in the list.</p>
"No caption (default value) has been specified for your button field."	<p>Every Button has to have a Default Value that is used as the label or caption on the button.</p> <p>Edit the form and enter a word in the Default Value field on the Advanced Field Properties screen of the Button field.</p> <p>Note that the form will still freeze in this case, because you can change Advanced Field Properties after the form is frozen.</p>
"The field name you have chosen is reserved by Pendragon Forms."	<p>Certain words cannot be used as field names: UnitID, RecordID, UserName, and TimeStamp are already used by Pendragon Forms.</p>
"Duplicate field name found – field names must be unique. Continue freezing form?"	<p>The database table that is created to store records associated with your form must have unique database column names. Because the database column names are based on the field names, if you have two fields with identical names, the database column names will be identical.</p>

Continued

Continued

Error Message Generated When Attempting to Freeze Form Design	Solution
"The default value does not match any item in the popup list."	<p>If you allow Pendragon Forms to continue freezing the form, an alternate database column name will be assigned to one of the identical fields. However, the field name that you see on the handheld will remain the same in both fields. You can edit the form and change one of the field names even after the form is frozen.</p> <p>You can default a Popup List to only one of the items in the list.</p>
"The default value for a checkbox must be either Y or N."	<p>Edit the form and change the Default Value in the Advanced Field Properties window to match the name of one of the items in the Popup List.</p> <p>To create a Default Value in a Yes/No Checkbox, use Y for Yes and N for No.</p> <p>Edit the form and change the Default Value in the Advanced Field Properties window accordingly.</p>
"The default value for an Option 1-5 field must be in the range 1-5."	<p>To create a Default Value in an Option 1 of 5 field, you can only use one of the numbers from 1 to 5.</p> <p>Edit the form and change the Default Value in the Advanced Field Properties window accordingly.</p>
"You created a required field with an invalid field type. Change the field type or make the field optional."	<p>Certain fields cannot be made required. These are Read-Only Text, Section fields, Jump Popup, Multi-Selection List, Subform, and Single Subform.</p> <p>Edit the field and uncheck the Required check box in the Advanced Field Properties window.</p>

HotSync Log Error Messages

A HotSync Log is generated every time you synchronize. At the end of the HotSync process, you can right-click the HotSync Manager icon and choose View Log to view the HotSync Log. If the synchronization was successful, you will see the word OK listed next to the name of each application that synchronized.

If a conduit generates an error message during the HotSync process, a HotSync Problem dialog box will appear at the end of the synchronization. Figure A-5 shows the HotSync Problem dialog box.



Figure A-5: The HotSync Problem dialog box

To see what the error message is, click the View Log button. The HotSync Log will be displayed, as shown in Figure A-6.

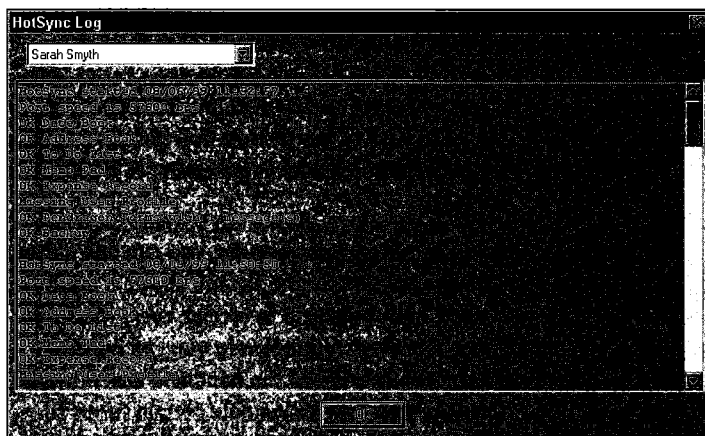


Figure A-6: The HotSync Log

Error messages that are generated by the Pendragon Forms conduit are shown in the following table.

**Pendragon Forms Conduit
Error Message**

"Missing User Profile"

Solution

For you to synchronize Pendragon Forms, your handheld user name must be listed as an active user in the User List in the Pendragon Forms Manager. The handheld user name must also be a member of a User Group.

To find out what your handheld user name is, tap the HotSync icon on the Palm device; the handheld user name will appear in the upper-right corner of the handheld screen.

Open the Pendragon Forms Manager and click the Users button. Enter your handheld user name exactly as it appears on the Palm device. Check the Active box to make yourself an active user. If you have a multiuser license for Pendragon Forms, enter all the handheld user names in the User List and make all the handheld users active.

Close the User List and then click the Groups button. If you are using Pendragon Forms with just one Palm device, click the Edit Members button next to the Default User Group. Add your handheld user name as a member of the Default group.

In a multiuser environment, you can add all the handheld users to the Default Group if you want each handheld to receive every form that is distributed. Alternatively, you can create a User Group and add the handheld users to your group, and then assign only certain forms to that group.

"License Count Exceeded. You must purchase more licenses."

Pendragon Forms ships as a single-user license. If you synchronize a second Palm device, this error message will be generated and the second handheld will replace the first as the active user in the User List.

**Pendragon Forms Conduit
Error Message****Solution**

If you have a multiuser license for Pendragon Forms and you are still getting this error message, make sure that the correct multiuser code has been entered. Click the Options button in the Forms Manager, and verify the multiuser code. Also click the Users button in the Forms Manager and verify that you have entered each handheld user name as the name appears on the handheld. (Tap the HotSync icon on the handheld to see the user name of the device.) Check the Active box to make each user active.

In a multiuser scenario you will also need to click the Groups button and assign each handheld user to a User Group.

"Unable to sync via DAO;-- ODBC connection to C:\Program Files\Forms3\Forms3.mdb failed."

On the handheld, a password has been entered where a password is not required.

On the handheld, tap the Forms icon, then tap the handheld Menu button and select the Login option (or use the Graffiti shortcut /O). If you have entered a user name and password, delete these items.

The user name and password here are used if the Pendragon Forms database has been password protected. If you have not password-protected the Pendragon Forms database, you do not need an Access user name and password on the handheld.

"Unable to remove form because it contains unsynced data."

This message might appear if a form is scheduled to be removed from the handheld, but there is a new or modified record that was unable to be synchronized. Because the record has not been backed up on the PC, the form cannot be removed.

Continued

Continued

**Pendragon Forms Conduit
Error Message**

Solution

<p>"Unable to append record – Index or primary key can't contain a null value."</p>	<p>On the handheld, tap the name of the form and then tap the Review button. As shown in Figure A-7, records that have not been sent to the PC will be marked with an arrow. Review any such records to determine why the record did not synchronize. The record may have a primary key field missing, a required field missing, or a primary key that conflicts with an existing record in the database. Modify the record and then retry the HotSync data transfer.</p>
<p>"Unable to append record – The changes you requested to the table were not successful because they would create duplicate values in the index, primary key, or relationship."</p>	<p>A primary key field has been left blank on the handheld. Edit the record on the handheld and then retry the synchronization.</p> <p>Microsoft Access generates this error if the primary key of a new record on the handheld matches the primary key of an existing record in the database.</p>
<p>"Unable to install the form design."</p>	<p>On the handheld, tap the name of the form and tap the Review button. Records that have not been synchronized will be marked with an arrow, as shown in Figure A-7. Review the records and modify the primary key field(s) so that they do not match the primary key of existing records in the database. Then retry synchronizing.</p> <p>This error message might appear if you import a form design and you omit to distribute the form, but you assign the form to a user group.</p> <p>Try redistributing the form and then performing a HotSync data transfer again.</p>
<p>"Unable to open database: Couldn't use database; file already in use."</p>	<p>This error message can appear if you are linking to an external database that has been opened exclusively by another user.</p>

Pendragon Forms Conduit Error Message

Solution

When one is opening a Microsoft Access database, there is an Exclusive option that allows the database to be used only by the person opening the database. If an external database has been opened exclusively, and a handheld user attempts to perform a HotSync data transfer, the database will be locked.

The solution is to close the external database and retry the HotSync data transfer.

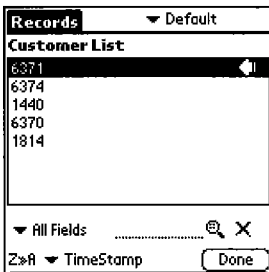


Figure A-7: Records that cannot be sent to the PC are marked with an arrow on the handheld.

Problems on the Handheld

Refer to the following table for the types of problems that may occur on the handheld.

Error Message or Problem on Handheld

Solution

A new record on the handheld does not appear to upload to the PC during synchronization.

(a) There could be a problem with the record that is causing it not to synchronize.

Continued

Continued

**Error Message or Problem
on Handheld**

Solution

Tap the name of the form, and then tap the Review button. Records that have not been uploaded to the PC will be marked with an arrow, as shown in Figure A-7.

Review the record(s). Check if there are primary key fields that match an existing record in the database, or if there are required fields that have not been filled in. Modify the record accordingly, and then reattempt a HotSync data transfer.

(b) The form could be missing from the PC. Refer to Chapter 10, "Managing Form Designs," for information on how you can retrieve a form design that has been deleted from the database on the PC.

Error message "Lookup failed. List not found or out of memory."

This message almost always occurs if a Lookup List has not been sent to the handheld. On very rare occasions, the error occurs if the handheld does not have enough free memory to store the Lookup List.

In the Forms Manager, click the name of the form and click the Edit button to open the Form Designer window. Display the Lookup List field, and check that a Lookup List has been selected in the Lookup List to Display field. If no Lookup List has been selected, or if a nonexistent list has been selected, reselect a Lookup List. Close the Form Designer window.

In the Forms Manager, click the Lookups button to view the Lookup Editor window. Check the following:

(a) The Lookup List that was referenced in the form actually exists. If it does not exist, create a Lookup List of the same name.

**Error Message or Problem
on Handheld**
Solution

	<p>(b) Display the Lookup List in the Lookup Editor window. The check box Make This Lookup Available on PalmPilot must be checked.</p> <p>(c) If you do not have a list of items in the second column – the LookupValue column – make sure that the check box Store LookupValues in the Lookup Field is <i>not</i> checked.</p> <p>Redistribute the form design and perform a HotSync data transfer. Check if the Lookup List now appears in your form.</p>
<p>I set a Data Persistence option to keep the records on the Palm for a certain length of time, and now I want to change that option.</p>	<p>On the PC, click the name of the form, and then click the Properties button to view the Form Properties window. Set the new Data Persistence option, then redistribute the form and perform a HotSync data transfer. The new Data Persistence options will take effect after synchronization.</p>
<p>Error message: "Missing Response (Required)"</p>	<p>A Required field has been left blank. Edit the field and enter a value.</p>
<p>Error message: "Value not allowed. (Number not allowed in range.)"</p>	<p>A Numeric range has been specified in a Numeric field. Edit the field to contain a number within the range.</p>
<p>Error message: "Primary Key Conflict. You must enter unique values in the primary key fields."</p>	<p>Each record must have a unique primary key in order to identify individual records on the handheld. This message appears if the primary key in the current record matches the primary key in another record, and you are trying to exit the current record.</p> <p>Edit the primary key field(s) of the current record so that they are different from existing records. Primary key fields have a key icon next to them in Record View.</p>

Continued

Continued

**Error Message or Problem
on Handheld****Solution**

A form cannot be deleted from the handheld.

If you delete a form from the handheld, and the form returns the next time you synchronize, try removing the form from the User Group to which you belong. Click the Groups button in the Forms Manager, and then edit the User Group of which your handheld user is a member. If you have not created a User Group, check the Default group.

If you are trying to manually delete a form from the handheld and you receive a message that the form cannot be deleted, the Advanced Form Property of Disable Form Deletion may be set. On the PC, click the form and then click the Properties button. Click the Advanced Form Properties button and uncheck the Disable Form Deletion check box. Redistribute the form and synchronize to send the updated form design to the handheld. (Note: if you have accidentally deleted the form design from the PC, you will not be able to change the Advanced Form Property of Disable Form Deletion. See Chapter 10, "Managing Form Designs", for instructions on how to attempt to retrieve a deleted form design.)

New records cannot be added on the handheld, or existing records cannot be modified.

In the Forms Manager, click the name of the form, and then click the Properties button. In the Access Rights section of the Form Properties window, see if the handheld rights have been set for No Additions or No Updates. Uncheck these boxes if necessary, and then redistribute the form and synchronize.

Problems with Scripts

When you write a script, click the Check Script button on the Script tab of the Advanced Field Properties window to check that your script is valid.

If you forget to check your scripts, you may see compilation errors on the handheld when the program tries to compile an invalid script. For example, you may have an `if . . . then . . . endif` statement that is missing the word `endif`.

When a compilation error occurs on the handheld, an error message will be displayed, and typically the script that is causing the problem is also displayed. You can correct the script on the PC and then redistribute the form to the handheld. Another type of error that can occur on the handheld is a runtime error. Division by zero causes a runtime error. For example, if a `calculate` script divides Field 1 by Field 2, and the handheld user enters a value in Field 1, the `calculate` script will run when the user leaves Field 1. Because nothing has been entered into Field 2 yet, the calculation will result in a division by zero error. The solution in this case is to use an `if . . . then . . . endif` statement so that the `calculate` script runs only when Field 2 is not zero. For example:

```
calculate:
if $2 = null then
  return
endif
answer = $1 / $2
```

Another type of scripting error that can occur is an error in the logical flow of the script. This type of error occurs if, for example, your script says `goto 5` but you meant it to say `goto 6`. The script will compile and run flawlessly, but it will not behave as you originally intended. The only way to catch this type of error is to go through the form field by field, triggering every script to see if the form behaves correctly. Correct any errors and redistribute the form to the handheld.

The following table outlines the errors that the script checker can detect.

Script Checker Error Message	Solution
"Event label expected."	Every script must begin with an event label that determines when the script will run. For example: <pre>calculate:</pre> If your script is missing an event label, add one at the start of the script. See Chapter 11 for a list of event labels.
"Conditional operator expected."	An <code>if..then...endif</code> statement is missing an operator. For example, <pre>if answer 5 then is incorrect if answer > 5 then is correct.</pre>

Continued

Continued

Script Checker Error Message	Solution
"Conditional operand missing."	An if...then...endif statement is missing an operand. For example, if = null then is incorrect if answer = null then is correct.
"THEN expected."	An if...then...endif statement is missing the word then.
"ENDIF without IF."	An if...then...endif statement is missing the word if.
"IF THEN statements cannot be nested."	Typically means that an if...then...endif statement is missing the word endif.
"\$ operator found without argument."	A number must follow the \$ operator in order to reference the value in a field. For example, \$5 refers to the value in Field 5.
"\$ operator requires a valid field number or identifier."	A \$ followed by a number references the value in a field. Because a form can only have up to 250 fields, the number that follows a \$ in a script has to be within the range 1-250.
"HIDE, SHOW, REQUIRE, OPTIONAL, READONLY, READWRITE and GOTO require a field number or identifier."	The scripting statements mentioned in this error message have to be followed by a field number. For example: hide 15
"FROM requires a field number or identifier."	A field number must follow the word from. For example: require from 5 to 10
"TO expected."	If you are referencing a range of field numbers, the words from and to are needed to specify the range. For example: show from 7 - 10 is incorrect. show from 7 to 10 is correct.
"= expected"	The words answer and result must always be followed by an equal sign. For example: answer 42 is incorrect. answer = 42 is correct.

Script Checker Error Message	Solution
"Missing value for LOOKUP statement."	The item that you are looking up must follow the word lookup. For example: lookup within "Prices" is incorrect. lookup \$4 within "Prices" is correct.
"WITHIN expected after LOOKUP argument."	The word within is missing from a lookup within statement. For example: lookup \$2 "Parts List" is incorrect. lookup \$2 within "Parts List" is correct.
"Missing lookup list name in LOOKUP statement."	The name of the Lookup List that you want to reference in the script is missing. For example: lookup \$18 within in incorrect. lookup \$18 within "Employees" is correct.
"Assignment operand missing."	An assignment statement must be followed by a value. For example: result = is incorrect. result = 200 is correct.
"COUNT statement requires a valid field number or identifier."	A field number is missing from a count statement. For example: count is incorrect. count 7 is correct.
"ANSWER or QUESTION expected after FONT."	The font statement must be followed by the word answer or question. For example: font bold is incorrect. font answer bold is correct.
"NORMAL, BOLD or LARGE expected for FONT."	The font statement must include the words normal, bold or large. For example: font question is incorrect. font question large is correct.

Continued

Continued

Script Checker Error Message	Solution
"PRINT statement expects port to be SERIAL or IR."	A print statement is missing the word <code>serial</code> or the word <code>IR</code> after the word <code>print</code> .

Problems with Bar Codes

With an SPT-1500, if you try to scan a bar code into a Text field and nothing happens, or if the bar code does not scan completely, refer to the following table.

Item to Check	Solution
Is the scanner enabled, and is the type of bar code that you are scanning enabled?	On the SPT 1500, tap the Applications button, and then tap the Diag (Diagnostics) icon. Select the Bar Code Scanner option, and then tap the down arrow to view screen 2. Ensure that the scanner is enabled, and that the bar code symbology that you are scanning is also enabled.
Is there a script in your form that is altering which bar code can be scanned, or altering how the scanned bar code is displayed?	If a script sets the bar code scanner, for example to scan only a certain bar code symbology, or to strip out leading and trailing characters of a bar code, then the settings will be global and will affect all fields unless another script resets the scanner.
Is the SPT 1500 being held too close to the bar code?	Sometimes, if you move the handheld farther away from the bar code, the scanner beam widens and can read the bar code.
Is the surface material of the bar code shiny?	If so, you may have to hold the SPT 1500 at an angle to the bar code surface for the scan to register.
Is the battery power low?	The SPT 1500 may not be able to scan bar codes if the battery power is low.

Appendix B

Scripting Syntax

CHAPTER 11, “Using Scripts,” provides examples of the types of scripts that can be used in Pendragon Forms. The syntax of the scripting language is provided here.

The text in a script uses white space to separate tokens – that is, symbols or words. White space includes one or more of the following characters: Carriage return (Enter), Line feed, TAB, space. The only tokens that do not need white space separators are the following operator symbols:

= & | ! # ^ % / * + - @

Legend

::= means definition

{x} means one or more of x

[x] means x is optional

x | y means x or y

UPPERCASE means literal keyword

Normal text means defined item

Syntax

script ::= event-handler {event-handler}

event-handler ::= event-name [statements]

statements ::= {statement}

event-name ::= OPEN: | CLICK: | CALCULATE: | INITIALIZE: | ENTER: |

EXIT: | SELECT: | VALIDATE:

statement ::= conditional-statement | action-statement

conditional-statement ::= IF condition THEN {action-statement} ENDIF

condition ::= value conditional-operator value

value ::= field-reference | constant | variable | function

constant ::= literal | NULL

literal ::= word | "non-alpha-word"

word ::= {alpha-character} [number]

number ::= {numeric-character}

alpha-character ::= A-Z | a-z

numeric-character ::= 0-9

```
non-alpha-word ::= {character}
character ::= any character except " | escape-sequence
escape-sequence ::= \ character

field-reference ::= ${numeric-character}

variable ::= ANSWER | RESULT

function ::= NOW

conditional-operator ::= > | < | = | <= | >= | <> | # | IS

action-statement ::= goto-statement | attribute-statement |
assignment-statement | query-statement | string-statement |
invalidate-statement | scanner-statement | other-statement

goto-statement ::= GOTO number

attribute-statement ::= attribute-statement-core number |
attribute-statement-core FROM number TO number

attribute-statement-core ::= HIDE | SHOW | REQUIRE | OPTIONAL |
READONLY | READWRITE

other-statement ::= FONT region font-type | LOOKUP

region ::= QUESTION | ANSWER
font-type ::= NORMAL | BOLD | LARGE

query-statement ::= lookup-statement | sum-statement | count-
statement

lookup-statement ::= lookup-type value WITHIN form-name
form-name ::= value
lookup-type ::= LOOKUP | REVERSELOOKUP

sum-statement ::= sum-type form-name column
column ::= value
sum-type ::= FORMSUM | SUBFORMSUM

count-statement ::= count-type column
count-type ::= COUNT | FILTERCOUNT

string-statement ::= LEFT string value | RIGHT string value |
```

MID string value value
string ::= value

invalidate-statement ::= INVALIDATE message
message ::= value

other-statement ::= control-statement | transmit-statement |
print-statement | MSGBOX message | GETADDRESS | BEEP

control-statement ::= ABORTFORM | RETURN | ENDFORM

transmit-statement ::= TRANSMIT transmit-mode address
transmit-mode ::= PALMNET | MULTIMAIL
address ::= value

print-statement ::= PRINT print-mode
print-mode ::= IR | SERIAL

assignment-statement ::= ASSIGN value | variable = expression |
field-reference = expression

expression ::= unary-operator value | value binary-operator value

unary-operator ::= LENGTH | - | INTEGER
binary-operator ::= + | - | / | * | % | # | &

scanner-statement ::= SCANNER scanner-command
scanner-command ::= scanner-action symbology | ENABLE ALL | DISABLE
ALL | conversion-type ENABLE | conversion-type DISABLE

scanner-action ::= ENABLE | DISABLE | CHECKDIGIT | SYSTEM_CHARACTER
| SYSTEM_CHARACTER_COUNTRY_CODE | NO_PREAMBLE | NO_CHECKDIGIT
conversion-type ::= UPCETOUPCA | UPCE1TOUPCA | CODE39TOCODE32 |
EAN8TOEAN13 | I2OF5TOEAN13

Appendix C

What's on the CD-ROM?

THE CD-ROM included with this book contains evaluation versions of the software products listed in the following table.

Product Name	Manufacturer	Contact Info
Pendragon Forms (version 3.0.12)	Pendragon Software Corporation	www.pendragon-software.com
Bachmann Print Manager (version 2.0)	Bachmann Software	www.bachmannsoftware.com
WaveSync synchronization server (version 1.5)	WaveWare Communications Inc.	www.waveware.net
MultiMail PRO (version 3.0)	Actual Software	www.actualsoft.com

Pendragon Forms

The evaluation version of Pendragon Forms is valid for 30 days from the date of installation.

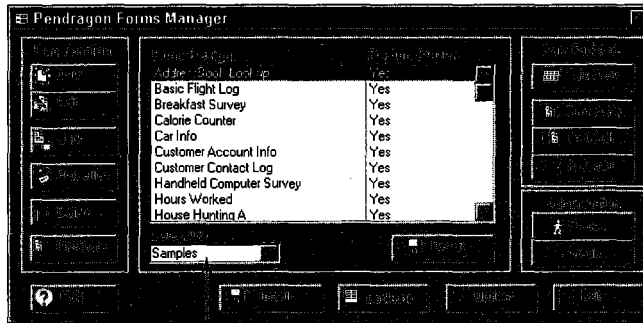


Installation instructions for Pendragon Forms can be found in Chapter 1, "What Is Pendragon Forms?"

Once you install Pendragon Forms, the Forms Manager database on your PC is preloaded with sample forms that are referenced throughout the book. You can send any of the sample forms to your Palm organizer.

To open the Pendragon Forms Manager on your PC, click Start → Programs → Pendragon Forms 3.0 → Pendragon Forms Manager.

Figure C-1 shows the Pendragon Forms Manager window. To view the sample forms, select the Samples category, as shown in Figure C-1.



Category field

Figure C-1: The Pendragon Forms Manager window

The sample forms that are referenced in each chapter are listed below.

CHAPTER 2: CREATING A FORM

Sample Form(s)	Features
Simple Expense Report	A small form used to illustrate different field types. This form is unfrozen and can be used to demonstrate freezing a form.

CHAPTER 3: ENTERING DATA ON THE PALM ORGANIZER

Sample Form(s)	Features
Handheld Computer Survey	Compare entering records for this form with AutoNavigate switched on and then with AutoNavigate switched off.
Travel Checklist	Compare entering records for this form in Field View versus Record View.

CHAPTER 5: FIELD TYPES

Sample Form(s)	Features
Project Tracker	A Lookup List.
Calorie Counter	A Cascading Lookup.
Customer Account Info	A reference form used with Customer Contact Log form.
Customer Contact Log	Performs a lookup to the Customer Account Info form.
House Hunting A	A form without Section fields.
House Hunting B	A form with Section fields used to separate categories of the form.
House Hunting C	A form with a Jump Popup field and Section fields for navigating through the form.
Patient Info (Parent)	A parent form.
Patient Visit Log (Subform)	A subform.
Insurance Review	A parent form for use with two single subforms.
Life Info	A single subform.
Car Info	A single subform.
Package Delivery	A form with a Signature field.
Address Book Lookup	A form with a Button field to perform a lookup to the built-in Address Book application.
Tasks To Do	A form with Completion Checkbox field.

CHAPTER 9: MANAGING DATA ON THE PC

Sample Form(s)	Features
Work Order Dispatch	A work order form that can be used to enter records on the PC that are sent to the handheld during synchronization.

CHAPTER 11: USING SCRIPTS

Sample Form(s)	Features
Hours Worked	A script that adds multiple fields together.
Shopping Survey	Branching using a Jump Popup field and scripts to branch back to the Jump Popup.
Breakfast Survey	Branching scripts based on selections in a Popup List field and a Yes/No Checkbox field.
Address Book Lookup	A Button field with a Click event script to perform lookup to a built-in Address Book application.
Vacation Cost Estimate	A button for performing a calculation.
Basic Flight Log	A button for adding across records on a form.
Simple Order-Taking	A Lookup . . . Within script.
Room-by-Room Inventory	A repeating form.

CHAPTER 12: USING BAR CODES

Sample Form(s)	Features
One Field Scan	A repeating form for scanning individual bar codes.
Room-by-Room Inventory	A repeating form with a script to enter a room number followed by repeated bar code scans.

Bachmann Print Manager

An evaluation version of Bachmann Print Manager software can be installed on your Palm organizer using the Palm Install Tool software that ships with the Palm device.

1. On the CD-ROM, select to install Bachmann Print Manager.
2. An Extract Print Manager Files window appears. Select to install to the default directory C:\PrintMgr or choose your own directory. Click the Next button.

3. The files will be extracted, and an Extraction Complete window will be displayed. Click the Finish button.
4. Microsoft Word will open to display the user documentation for Bachmann Print Manager. You can print the documentation or reference the Word file at a later time in the directory where you installed Bachmann Print Manager. Close Microsoft Word when you have finished viewing the documentation.
5. You will return to the CD-ROM splash screen. Click the Exit button.
6. To install Bachmann Print Manager on your handheld, click Start → Programs → Palm Desktop → Palm Install Tool.
7. Click the Add button in the Palm Install Tool window. An Open window is displayed. Select to view the C:\PrintMgr\PageMgr(Demo)\Bin folder.
8. There are two .prc files to install: IrMgr.prc and PageMgr(Demo).prc. Click on each file in turn, and then click Open. The file will be added to the list of files to install on the handheld. Click the Add button in the Palm Install Tool window to select the second of the two .prc files.
9. Perform a HotSync data transfer to install the two .prc files.

Once Bachmann Print Manager has been installed on the handheld, you can print records from Pendragon Forms via the infrared port of the handheld to a printer with an infrared port.



See Chapter 3, "Entering Data on the Palm Organizer," for additional information on printing via infrared. Also see Chapter 11, "Using Scripts," for information on creating a Print Button within a form.

WaveSync Synchronization Server

The WaveSync synchronization server is used for synchronizing multiple handheld devices wirelessly or via a network.



See Chapter 15, "Planning a Multi-User Installation," for information on when you might need to use the WaveSync server instead of the HotSync Manager.

WaveSync consists of server software and proxy software. The server software has to be installed on a Windows NT 4.0 PC with Service Pack 3 or later. The proxy software can be installed on any Windows 95, Windows 98, or Windows NT PC with which a handheld will synchronize. To synchronize over a network, your network must support the TCP/IP protocol.

The evaluation version of the WaveSync server supports two users using a maximum of two conduits for four hours at a time. The full product supports a 24-hour Windows NT service for a licensed number of handheld users.

Installing the WaveSync Server

To install WaveSync, insert the CD-ROM into the CD-ROM drive and select the option to install WaveSync. Follow the on-screen prompts during the installation process.

Once WaveSync is installed, click Start → Programs → WaveSync → WaveSync Manager. You will be prompted to select a server, as shown in Figure C-2. Select a server name and then click Connect.

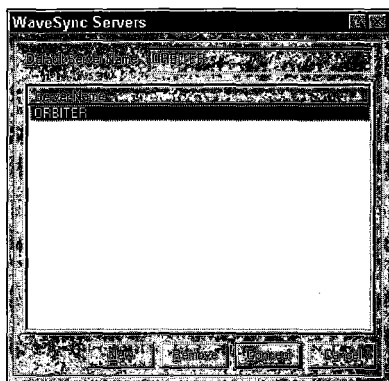


Figure C-2: Select a server.

The WaveSync Manager window opens. Figure C-3 displays the WaveSync Manager.

In order to work with a third-party application such as Pendragon Forms, WaveSync needs the specifications for what the Pendragon Forms conduit file is called and where it is located. This information is stored in what is called a SyncMap.

Pendragon Forms ships with a ready-made SyncMap in a file called Forms3.wsm. To import the Pendragon Forms SyncMap into WaveSync, click File → Import SyncMap Wizard. Choose to import the file Forms3.wsm from the C:\Program Files\Forms3 folder.

After importing the Pendragon Forms SyncMap, click the SyncMap button in the WaveSync Manager window. Figure C-4 shows that the Pendragon Forms conduit will be listed as one of the conduits that can synchronize via WaveSync.

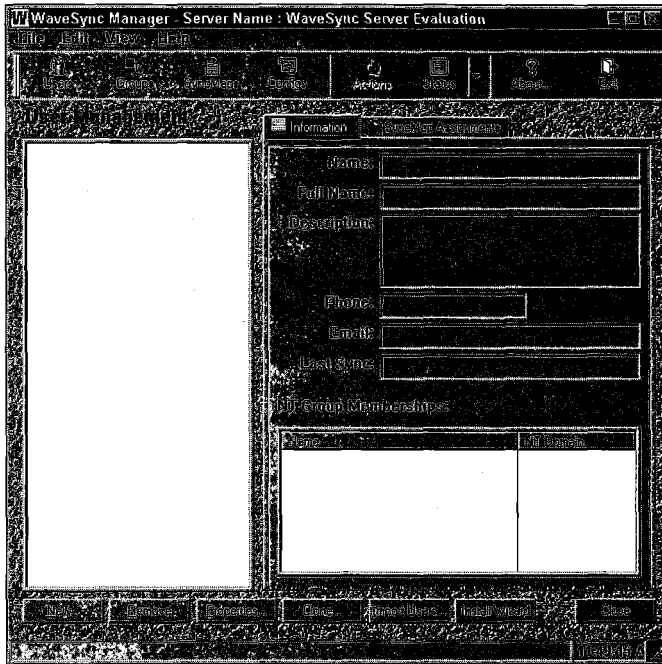


Figure C-3: The WaveSync Manager

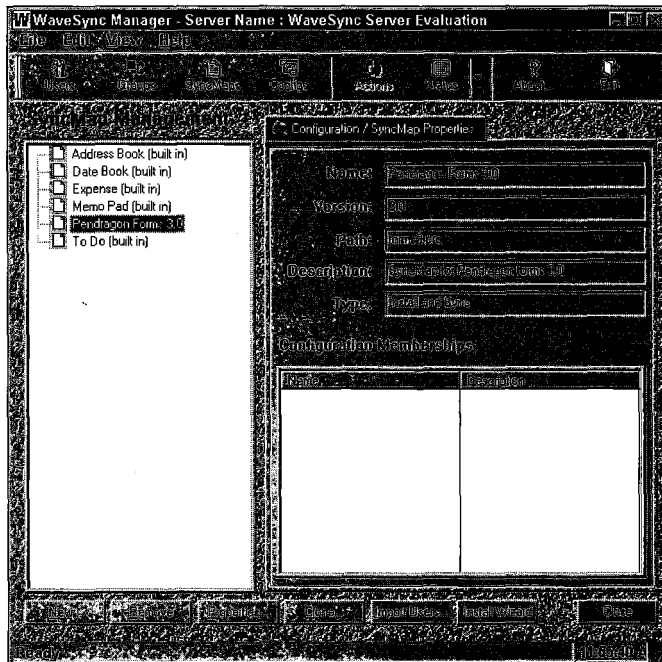


Figure C-4: The list of SyncMaps

Click the Users button in the WaveSync Manager. To create a new user, click the New button. Figure C-5 shows the user profile window.

- ◆ The Name field is the user's Windows NT login name.
- ◆ The Full Name field is the user's actual name.
- ◆ The Password is the user's password to log onto Windows NT.

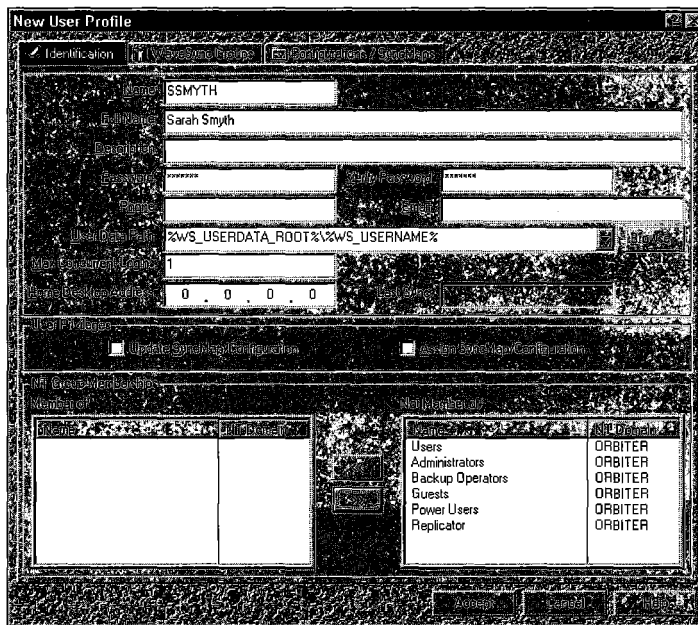


Figure C-5: The User Profile

Click the Configurations/SyncMaps tab to select the conduits that the user will be allowed to synchronize via WaveSync. (Note that for the evaluation version you can select only two conduits.) Figure C-6 shows the Pendragon Forms conduit selected for a user.

When you have finished creating a user, the user will appear in the list of WaveSync users, as shown in Figure C-7.

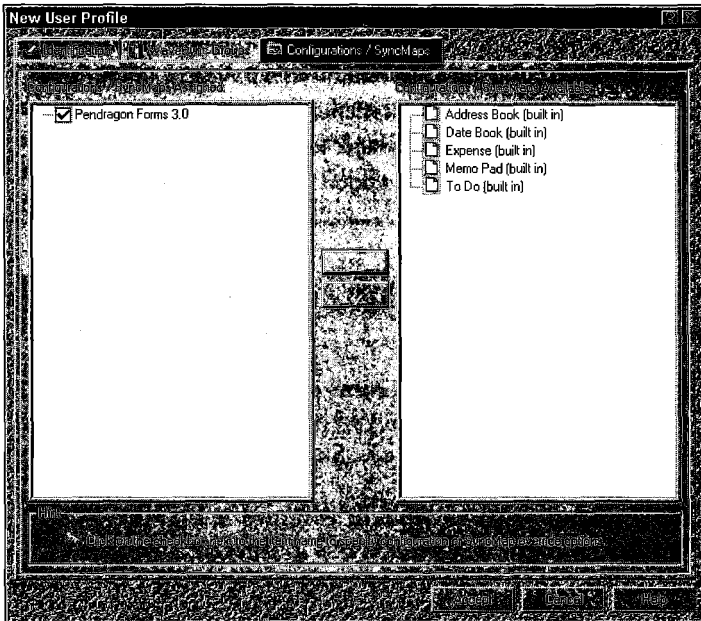


Figure C-6: Selecting conduits for a WaveSync user

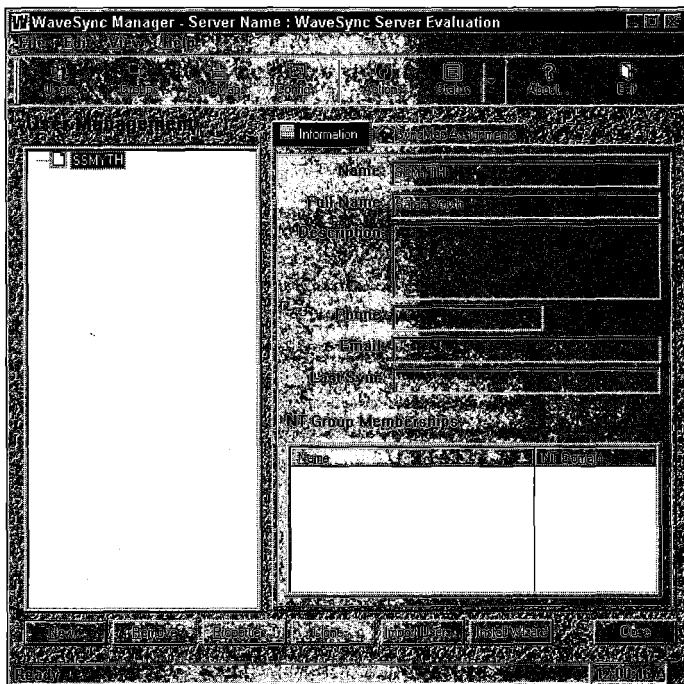


Figure C-7: The list of WaveSync users



Users of Pendragon Forms will need access to the Pendragon Forms database file (Forms3.mdb, or Forms32K.mdb if using Access 2000). Your Windows NT system administrator must manually adjust the folder access privileges with the Windows NT User Manager to allow the NT user accounts access to the directory in which the Pendragon Forms Database is located.

Preparing Pendragon Forms for Use with WaveSync

Before you can begin synchronizing Pendragon Forms using WaveSync, Pendragon Forms has to be installed on the server, and you will need to configure the Pendragon Forms conduit. On the server, click Start → Programs → Pendragon Forms 3.0 → Configuration Tool.

The Pendragon Forms Configuration Tool window is shown in Figure C-8. When using WaveSync, verify that the check box labeled Use a Separate Thread for ADO Access (Deactivate for WaveSync) is not checked.

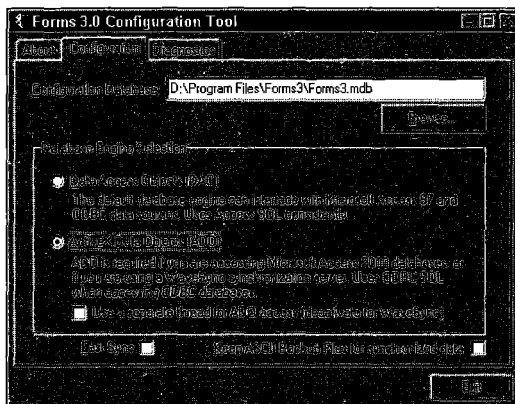


Figure C-8: Pendragon Forms Configuration Tool window

Installing the WaveSync Proxy

Once the WaveSync server has been installed, you can install the WaveSync proxy software on a workstation with which the handheld will synchronize. You can choose to install the WaveSync proxy on the same PC as the WaveSync server.

One of the benefits of using WaveSync is that you do not need to install Pendragon Forms on the workstation.

On the workstation PC, close the HotSync Manager. Then open Windows Explorer and access the WaveSync\Install\Desktop Proxy folder on the server. To

install the WaveSync proxy, double-click the Setup.exe program in the Desktop Proxy folder.

During the installation of the WaveSync proxy software, you will be asked if you want to migrate the Palm Desktop, as shown in Figure C-9. Migrating the Palm desktop means using WaveSync for synchronizing the built-in Palm applications such as the Address Book and Date Book. If you are just evaluating WaveSync at present, un-check the check box so that the Palm Desktop is not migrated at this time.

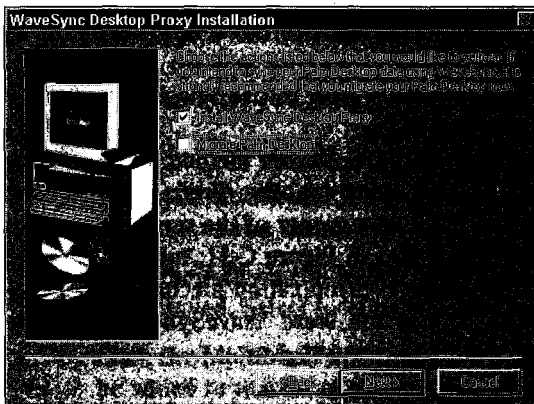


Figure C-9: For evaluation purposes, choose not to migrate the Palm Desktop.

After the WaveSync proxy software has been installed, right-click the WaveSync proxy icon in the Windows system tray and choose Install WaveSync Client. The WaveSync Client Installation window, shown in Figure C-10, is displayed.

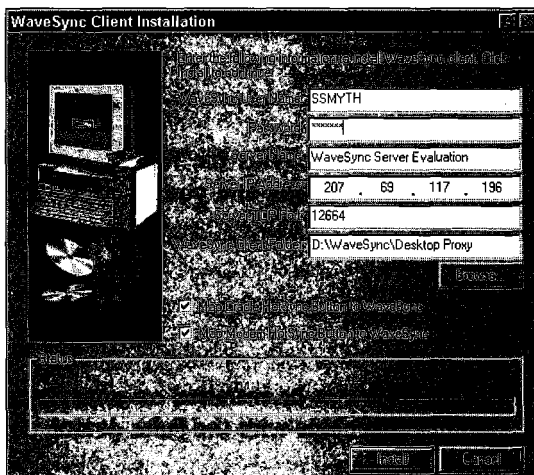


Figure C-10: WaveSync Client installation

Enter the user's Windows NT login name and password in the WaveSync Client Installation window.

To install WaveSync on the handheld, place the Palm device in the HotSync cradle and press the HotSync button on the cradle. Once WaveSync is installed on the handheld, you can synchronize Pendragon Forms using WaveSync.

Switching between WaveSync and the HotSync Manager

When WaveSync is installed on the handheld, it replaces the HotSync process as the default method of synchronization.

While you are evaluating WaveSync, you may want to use WaveSync at some times and use the HotSync Manager at other times.

To make the HotSync process the default synchronization method again, tap the Applications button on the handheld, and then tap the Prefs icon. In the Preferences screen, tap the arrow in the upper-right corner of the screen and select the Buttons menu. On the Buttons screen, shown in Figure C-11, tap the HotSync button. WaveSync will be listed as the default synchronization mechanism for cradle and modem. Tap the arrow to the left of the word WaveSync and choose HotSync from the list of options.

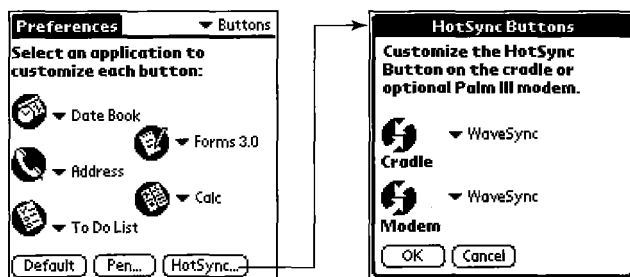


Figure C-11: Choosing a synchronization method on the handheld

If you choose to implement the full version of WaveSync, the WaveSync synchronization mechanism will replace the HotSync Manager and you will not need to switch between the two.

MultiMail PRO

An evaluation version of MultiMail PRO software can be installed on your Palm organizer using the Palm Install Tool software that ships with the Palm device.

1. On the CD-ROM, select to install MultiMail PRO.
2. An Extract MultiMail PRO Demo Files window appears. Click the Next button.
3. A Destination Directory window appears. Choose to install to the default directory C:\MultiMail, or select your own directory. Click the Next button.
4. The files will be extracted. Click the Finish button.
5. Windows Notepad will open to display the instructions for using MultiMail PRO. You can print the documentation, or refer to the README file in the directory where you installed MultiMail PRO. Close Windows Notepad after you have reviewed the documentation.
6. You will return to the CD-ROM splash screen. Click the Exit button.
7. To install MultiMail PRO on your handheld, click Start → Programs → Palm Desktop → Palm Install Tool.
8. Click the Add button in the Palm Install Tool window. An Open window is displayed. Select to view the C:\MultiMail folder.
9. Click the MultiMailPROIII.prc file, and then click Open. The file will be added to the list of files to install on the handheld.
10. Perform a HotSync data transfer to install the .prc file.

After installing MultiMail PRO, you can create a Button field in Pendragon Forms to send the contents of a record via e-mail.



See Chapter 11, "Using Scripts," for information on creating an E-Mail Button within a form.

Glossary

Access Rights Options that determine whether records can be added, updated, or deleted on the handheld.

Additional Download Criteria An Advanced Field Property containing criteria that must be met before a given record is transferred to the handheld during synchronization.

Advanced Field Properties Attributes that can be set on a field-by-field basis.

Advanced Form Properties Attributes that affect the way a form displays on the handheld, and that affect which records are sent to the handheld.

Autodefault A property of a field that causes it to remember the last value entered. This feature is often used when the same value will be recorded in a field of several records in a row.

AutoNavigate An option that speeds data entry on the handheld by automatically displaying dialog boxes, and automatically jumping to the next field when a selection is made.

AutoNumber A type of field in Microsoft Access that stores a unique sequential number in each record.

Bar Code Symbology The format used for encoding a bar code. Examples are UPC A, CODABAR, and CODE39.

Bi-Directional Synchronization Synchronization in which new records on the handheld are uploaded to the PC, and new records on the PC are downloaded to the handheld.

Cascading Lookup List A Lookup List field that displays a different list depending on the values selected in previous fields.

Column Name The name assigned to a column in a database table.

Completion Checkbox A special type of Yes/No Checkbox field. When a Completion Checkbox is marked Yes on the handheld, the record is removed from the handheld if the appropriate Data Persistence option has been set.

Conduit An extension to the HotSync Manager that enables an application on the handheld to synchronize with the PC during a HotSync data transfer.

CSV File Comma-separated variable format is a text file format in which each record is stored on its own line, and all fields are separated by commas.

Data Persistence Form property settings to determine how long records remain on the handheld.

Default User Group The User Group to which all forms designs are assigned when the forms are distributed. In a single-user environment, the handheld user is a member of the Default User Group.

Default Value The value that will be assigned to a field if the user makes no alternative selection.

Display Key The column designated for display when one is reviewing lists of records on the handheld. The default Display Key is the first field on the form. The Display Key can be changed by accessing the Advanced Field Properties screen.

Distributing a Form Marking a form as ready to be sent to handheld devices. Distributing a form adds the form to the Default User Group. If a form design is modified and the form is redistributed, the updated form design is sent to all handhelds in the User Group to which the form has been assigned.

Event Procedure A sequence of instructions triggered by an event. Events include opening a form, creating a new record, displaying a specific field, and closing a form.

Field An item of information on a form.

Field Mapping The process of associating fields on a form with columns in a database table.

Field Name A description for a field that the handheld user sees in order to know what type of information to enter in the field.

Field Type A property of a field that determines its function and the type of data it can hold – for example, Text, Numeric, Yes/No.

Field View A display mode on the handheld that shows the user one field of the form at a time.

Filter A set of criteria on the handheld that determines which records are visible on the handheld.

Form A template for collecting fields of information.

Form Designer Window A tool in the Pendragon Forms Manager used for designing forms.

Form ID A unique number that is based on the creation date and time of a form design. The Form ID is used in the name of the database table that is created when a form is frozen.

Form Properties Attributes that affect how long records can remain on the handheld, and whether the handheld user can create and update records on the handheld.

Forms List A list of form designs on the handheld. The Forms List is the first screen that you see when you tap the Forms icon on the handheld Applications screen.

Freezing a Form The process of creating a database table in the Pendragon Forms Manager database for storing records associated with a form. A form must be frozen before it can be sent to the handheld. Freezing a form makes some aspects of the form design unchangeable.

Handheld A generic name for personal digital assistant computers. In this book, *handheld* refers to the family of Palm Computing organizers.

Handheld User Name See Palm User Name.

HotSync Data Transfer The process of synchronizing the handheld with the PC. A HotSync can be local – via the HotSync cradle attached to the PC – or via modem.

HotSync Manager The software program that runs on the PC to perform the synchronization with the handheld. The HotSync Manager is part of the Palm Desktop software that ships with the Palm organizer hardware.

Linked Table In Microsoft Access, a table that stores its data in an external database.

Lookup List A field that presents the user with a list of options to choose from. The list can be modified after the form is frozen.

ODBC Open Database Connectivity is a universal method for accessing various database systems.

Palm Install Tool A program that is used to install programs onto the handheld. The Palm Install Tool is part of the Palm Desktop software that ships with the handheld.

Palm OS The operating system that runs on a Palm computer.

Palm User Name The user name that has been assigned to a handheld device. To see what the handheld user name is, tap the HotSync icon on the handheld. The Palm user name appears in the upper-right corner of the handheld screen.

Parent Form A form with a Subform List field that links to a related form.

Pendragon Forms Conduit The conduit used by the Pendragon Forms program to send form designs to the handheld and synchronize data between handheld and PC.

Pendragon Forms Manager A database on the PC that enables you to design forms for use on the handheld. The Forms Manager is in Microsoft Access format.

PFF File A file that contains the design of a form. Internally, .pff files are stored in text format.

PQA A Palm Query Application is a static Web page used for launching Web requests on a Palm VII.

Primary Key In a database table, a column or collection of columns that uniquely define a record. Also, an attribute of a field on a form that indicates it is part of a primary key.

Record The information created by filling out a single form.

Record View A display mode on the handheld that shows the form in a two-column format, the left-hand column containing field names and the right-hand column showing the responses to each field.

Script A sequence of instructions that may be used for controlling the display of fields, calculating values, or performing other actions. Scripts are composed of event procedures and are associated with specific fields.

Single Subform A subform that allows only a single subform record for each parent record.

Statement A single instruction that is part of a script.

Subform A form that is referenced by another form. Subforms inherit field values from parent forms when they are created.

Table Name The name assigned to a database table.

TimeStamp A field that records the date and time when a record was created. Although the TimeStamp is not visible on the handheld, it can be seen when a record is uploaded to the PC during synchronization.

User Group A list of handheld users and the forms that are assigned to those users.

User List The list of the handheld user names that can synchronize Pendragon Forms.

UserName A field that associates a record with the Palm User Name of a handheld device. Although the UserName is not visible on the handheld, it can be viewed when a record is uploaded to the PC during synchronization.

Validation The process of checking that field values entered by the user are consistent and/or within an allowed range of values.

Index

Symbols and Numerics

; (semicolon) Multi-Selection List item separator, 181

/ (slash) division operator, 224

&t (ampersand)

- concatenation operator, 225
- Web address suffix, 351

> (arrow) greater than operator, 170, 227

< (arrow) less than operator, 170, 227

>= (arrow, equal sign) greater than or equal to operator, 227

<= (arrow, equal sign) less than or equal to operator, 227

<> (arrows) not equal operator, 170, 227

* (asterisk) multiplication operator, 224

@ (at symbol) cascading relationship indicator, 87

= (equal sign) equal to operator, 170, 227

% (percent sign) Modulo division operator, 224

+ (plus sign) addition operator, 224

? (question mark) CGI program name suffix, 351

(hatch) string matching operator, 224, 227

##USERNAME##, 171, 300-301

- (minus sign)

- change number sign operator, 223
- subtraction operator, 216-217, 224

[UserName] =, 171

{enable|disable} barcode, 271

3Com Corporation, 4. See also names of 3Com products

A

abortform, 228

Access. See *also* linking to external Access databases

- Access 97, 177
- Access 2000, 177
- Access database table view, 184
 - database file name, 15
 - Design View mode, 189-190
 - ISAM drivers, 320
 - maximum text field length, 133-134, 298
 - Print Preview mode, 191
 - reports, 187-193
 - storage location, 177, 344
 - split screen with Pendragon Forms, 187-188
- Access database table view, 184

access rights, 50-51, 299, 305, 314, 335-336

Address Book

- lookup buttons, 113, 251-252
- lookup Graffiti shortcut (/L), 69
- pasting to text fields, 69
- user prompt script, 230

Advanced Field Properties window, 118

Advanced Form Properties window, 155

- age calculation scripts, 244–245
 - All Users Group, 144
 - American Microsystems, 266
 - AND relational operator, 170
 - answer variable, 217, 222, 225
 - application development tool choice, 9–10, 348–349
 - Archive Date, 149, 169
 - arithmetic operations. *See* math operations
 - assign statement, 226
 - Auto-Execute, 164–165
 - Autodefault, 124–125, 262
 - AutoNavigate
 - described, 35–36
 - disabling, 159–160
 - AutoNumber field (Microsoft Access), 131, 132, 296–297
 - AutoRepeat, 164, 261–263
 - Average function, 42–43
- B**
- Bachmann Print Manager, 52, 234
 - Back button, hiding, 157–158
 - backups
 - data within a form, 200
 - database, 199
 - form design, 199
 - bar codes
 - checkdigits, 273
 - code conversion, 273
 - enabling/disabling scanner, 271
 - field types, 267
 - form design, 267–269
 - inventory use, 276–277
 - library circulation, 279–281
 - prefix options, 272, 275
 - as Primary Key, 278
 - record retrieval using, 277–278
 - Record View, 268
 - resetting scanner at field exit, 274
 - serial port-attached scanners, 268–269
 - SPT scanner input control, 270–273
 - suffix options, 273, 275
 - supported scanners, 265–266
 - sympology control, 270, 272
 - uniqueness, 278
 - UPCE to UPCA conversion, 274–275
 - battery life, 3–4
 - beep, 228
 - bi-directional synchronization, 7, 30
 - Button field
 - Address Book lookup, 113, 251–252
 - calculate buttons, 236, 252–254
 - click events, 111, 112, 220
 - data entry, 112
 - default value, 122
 - e-mail buttons, 256–257
 - maximum label length, 112
 - Palm VII transmit button, 349–353
 - print buttons, 254–255
 - scripting, 250–251
 - uses, 112, 114
- C**
- calculate events, 124, 216–217, 219, 220
 - carrying values forward, 124–125, 262

cascading relationship, 86–87

CD

- Bachmann Print Manager, 52
- installing Pendragon Forms, 10–11
- sample forms, viewing, 15
- system requirements, 12
- version information, 10
- WaveSync synchronization server, 341

CGI script, 351

Change Flags option, 46–47

character type, restricting entry, 135–136

checkboxes

- Completion Checkbox field, 114–116, 173, 182, 185, 289
- Option 1 of 5 field, 74–76, 181
- Time Checkbox field, 95–96
- Yes or No Checkbox field, 73–74, 182

click events, 111, 112, 220

Click View(Properties, 192

columns

- field display, 30
- names, 20, 119, 120
- sorting by, 174

comma-separated variable (CSV) files

creating in Excel, 198

exporting to, 193

importing from, 196–197

Completion Checkbox field. *See also*

Yes or No Checkbox field

data entry, 115–116

data entry on PC, 182

described, 114–115

limiting download using, 173

linked external databases, 289

uses, 116

work orders, 185

component interaction, 6–8

Copilot bar code scanner, 266, 270

count *field-number*, 229

counters, 241

CSV files. *See* comma-separated

variable (CSV) files

Currency field

data entry, 73

data entry on PC, 181

default value, 121

introduced, 17

maximum length, 72

scripts, 240

uses, 73

D

data entry. *See also* data entry,

controlling

adding new records, 25, 33–40

bar codes, 267–269

Button field, 112

carrying values forward, 124–125, 262

Completion Checkbox field,

115–116, 182

Currency field, 73

Date & Time field, 92, 182

Date Only field, 94, 182

exiting records, 26

Freeform Text fields, 68

Jump Popup field, 100

Continued

- data entry *continued*
 - Lookup List field, 83–84, 181
 - Lookup List option, 134–135
 - Multi-Selection List field, 78–79
 - Numeric fields, 71, 181
 - Option 1 of 5 field, 75, 181
 - PC data entry, 180–182
 - Popup List field, 77, 181
 - Section field, 98
 - Signature field, 111–112
 - Single Subform field, 108–109
 - subforms, 102–105
 - Time Checkbox field, 96, 182
 - Time field, 95, 182
 - Yes or No Checkbox field, 74, 182
- data entry, controlling
 - access rights, 50–51
 - integers, limiting to, 71, 133
 - maximum/minimum numeric values, 132–133
 - Required fields, 126–127, 160–161, 233
 - restricting record creation to parent form, 162–163
- data persistence
 - changing, 22
 - Completion Checkbox use, 114
 - default, 21
 - incomplete record options, 22, 150
 - keeping handheld records after synchronization, 22
 - limiting records on handheld, 62–63
 - linked external databases, 298–299, 312
 - subforms, 107
- datasheet view, 29–30, 178–179
- date
 - capturing, 92
 - date scripts, 241–246
 - limiting download by, 172
 - Palm Computing Platform limitations, 93
- Date & Time field
 - data entry, 92
 - data entry on PC, 182
 - default value, 92, 122, 123
 - initializing using scripts, 123
 - limits, 92
 - uses, 93
- Date Only field
 - data entry, 94
 - data entry on PC, 182
 - default value, 94, 122, 123
 - initializing using scripts, 123
 - introduced, 16
 - limits, 93
 - uses, 94
- dBase. See linking to external ODBC databases
- Delete button, hiding, 166–167
- DESC keyword, 174
- Design View mode, 189–190
- Detailed Form Design Report, 32
- developer Web site, 354
- development tool choice, 9–10, 348–349
- dialog boxes, creating, 233
- display, handheld
 - default, 25–26, 40
 - field numbers, 38–39
- Field View, 26–28, 34–36, 39–40

Record View, 26–28, 37–40
 subforms, 162–164
 time, 95
 Wide mode, 38

display, PC
 Access database table view, 184
 columns, 30
 datasheet view, 29–30, 178–179
 Form View, 183
 linked external databases, 308
 Query view, 183–184
 rows, 30

Display Key field
 described, 18, 125–126
 subforms, 102, 104–105, 106

duplicate records, 129–131, 167–168

Dynasys Technologies, 266

E

e-mail scripts
 assigning to buttons, 256–257
 creating, 235

{enable|disable} barcode, 271

encryption, 344, 345

End button, hiding, 156–157

endform, 229

enter events, 220, 221, 271

event procedures
 calculate, 124, 216–217, 219, 220
 click, 111, 112, 220
 enter, 220, 221, 271
 exit, 220, 246, 271
 initialize, 94, 123, 220, 221
 open, 220, 221, 271
 select, 220, 221–222, 246
 validate, 220, 222, 232, 260–261

Excel. *See also* linking to external
 ODBC databases
 creating CSV files, 198
 primary key, 320
 reports, 193

Exclusive Lookup List field, 80–85,
 122, 181

exit events, 220, 246, 271

exiting Pendragon Forms, 26

Export Data window, 203

exporting. *See also* linking to external
 Access databases; linking to
 external ODBC databases
 database to ASCII, 193
 forms, 202–203, 207
 Lookup Lists, 204–205

F

Field Mapping Tool window, 302

Field View
 described, 26–28, 34–36
 setting as default, 160
 subforms, 102
 toggling from Record View, 39–40

fields. *See also* names of specific
 fields
 adding, 15
 Advanced Field Properties, 117–119
 column display, 30
 default value, 120–123, 124–125
 displaying field numbers, 38–39
 Do Not Upload to PC property,
 131–132
 first field, 18
 goto field number, 231

Continued

- fields *continued*
- hiding, 123, 231–232
 - linked external Access databases, 292–297, 301–306
 - linked external ODBC databases, 327–329
 - Lookup List option, 134–135
 - maximum length property, 68
 - maximum number, 15, 67
 - maximum number, extending
 - using subforms, 107–109
 - naming, 14, 28
 - navigation, 34–36, 38
 - non-printing, 132
 - Pattern option, 135–136
 - PC, updating on, 181–182
 - Read-Only attribute, 124
 - Required, 126–127, 160–161, 233
 - types, listed, 67–68
 - types, selecting, 14–16
 - unbound, 305, 306
 - validation, 160–161, 232, 260–261
 - viewing, 26–27
- filtercount *field-number*, 229
- filtering records, 44–45, 229
- font {answer|question}
 - {bold|normal|large}, 229
- Form Design Report, 32
- Form Designer window, 13–15
- form designs
 - backing up, 199
 - copying, 31–32, 120, 207–208
 - creating, 13–15
 - deleting, 49–50, 146–147
 - deleting accidentally, recovery, 165, 166, 202
 - deleting from user groups, 145
 - deletion disabling, 166
 - design tips, 58
 - Display Key field choice, 18
 - distributing, 24–25, 146, 307
 - editing, 19
 - editing frozen forms, 31, 118
 - exporting, 202–203
 - fields, adding, 14–16
 - freezing, 20–23, 119, 129
 - ID number, 199–200, 202, 208–209
 - importing, 208–209
 - naming, 14
 - organizing designs, 23
 - performance issues, 58–62
 - printing, 32
 - redistributing, 22
 - saving, 15
 - sending to handheld, 24–25, 146, 307
 - storage location, 177–178, 201
 - synchronizing, 24–25, 146, 202
 - synchronizing, switching off, 150–151
- Form ID#, 199–200, 202, 208–209
- Form Properties window, 21
- Form View, 183
- forms. *See* form designs
- Forms conduit, 6–8, 137, 138–139
- Forms List screen, 25
- Forms Manager, 6–8
- Forms program for Palm OS, 6–8
- Forms3: Database window, 321
- forms3a.dll file, 137
- forms3.dll file, 137
- forms3.mdb file, 177, 344

Forms3.prc program, 11
 forms32k.mdb file, 177, 344
 formsproA.mdb file, 206
 formsproB.mdb file, 206
 formspro.mdb file, 206
 formsum “*name-of-form*” *field-number*, 230, 253–254
 FoxPro. *See* linking to external ODBC databases
 Freeform Text field
 data entry, 69
 data entry on PC, 181
 introduced, 16
 Lookup List option, 134–135
 maximum length, 16, 69, 133–134
 Pattern option, 135–136
 restricting character type entered, 135–136
 uses, 70

G

getaddress, 230
 Global Positioning Satellite (GPS) systems, 69
 goto *field-number*, 231
 Graffiti (handwriting recognition system), 4
 Graffiti shortcuts
 Address Book lookup (/L), 69
 Toggle Change Flags (/M), 47
 Toggle Default View (/V), 40, 160

H

handheld, synchronizing with PC. *See* synchronization
 handheld user name, 142, 171

Handspring Visor Deluxe Palm OS device, 5
 Handspring Visor Palm OS device, 5
 handwriting recognition, 4
 Hidden fields, 123
 hide *field-number*, 231
 hide from *field-number* to *field-number*, 231
 HotSync. *See* synchronization

I

IBM Workpad, 5
 if . . . then . . . endif, 226
 impersonation, 343
 Import Data dialog box, 292
 Import Existing Form Data window, 208
 Import Existing Forms Database window, 206
 importing. *See also* linking to external
 Access databases; linking to external ODBC databases
 data from another form, 207–208
 data from ASCII file, 196–197
 data from previous versions, 205–207
 features overview, 205
 form designs, 208–209
 Lookup Lists, 209–211
 user lists/groups, 207
 Informix. *See* linking to external ODBC databases
 initialize events, 94, 123, 220, 221
 inserts on handheld, 150, 167–168
 Integer Advanced Field Property, 71, 133

integer *value*, 223
 integers, limiting entry to, 71, 133
 invalidate "message", 232
 inventory, 276–277
 ISAM drivers, 320

J

Jump Popup field. *See also* Popup List field
 branching scripts, 247
 data entry, 100
 described, 98–99
 maximum length, 100
 returning to using script, 100
 Section field correspondence, 100
 uses, 100

L

LANs. *See* Local Area Networks (LANs)
 left *value length*, 232
 Legal Terms box, 110
 length *value*, 223
 library circulation, 279–281
 licensing, 207, 333
 Link Tables window, 323
 linking to external Access databases
 access rights, 299, 305, 314
 Archive Date limitations, 169
 Autonumber fields, 287, 296–297
 column names, 120
 Completion Checkbox fields, 289
 Currency field truncation, 298
 data persistence options, 298–299, 312
 database table preparation, 286–289

 download criteria, 300–301
 field mapping, 301–306
 field names generated, 292
 field type, changing, 292–293
 field type compatibility, 293–295
 forms from external queries,
 creating, 314
 forms from external tables,
 creating, 290–292
 forms from external tables, editing,
 292–298
 handheld memory issues, 289
 ISAM drivers, 320
 linked table creation, 320–325
 linking to external table, 301–306
 Memo field conversion, 296
 Memo field truncation, 198
 Numeric fields, 296
 password protected databases, 320
 Primary Key, 129, 286–287,
 295–296, 305, 307
 RecordID issues, 288
 sending forms to handheld, 307
 sending records to/from handheld,
 298–301
 steps, 285
 subforms, 308–313
 synchronization, 298–301, 307,
 310–313, 315
 timestamp issues, 288
 unbound fields, 306
 UnitID issues, 288
 UserName issues, 288–289
 viewing data on PC, 308
 linking to external database query,
 313–315

linking to external ODBC databases

- Archive Date limitations, 169
- column names, 120
- data source creation, 318
- data source selection, 326
- field mapping, 327–329
- form creation, 326
- form editing, 327
- machine Data Source, 320, 321, 325–326
- mapping directly, 319, 325–326
- ODBC driver, 317–318
- passwords, 326, 330
- record control, 327
- security, 329–330
- steps, 319
- using linked table in Access, 318–325

Local Area Networks (LANs)

- Network Setup utility, 337–338
- speed, 339
- synchronization, 337–342

lookup ... within, 257–260

Lookup Editor window, 61

Lookup List field

- branching scripts, 248–249
- casading, 85–87
- creating, 80–83
- creating from ASCII files, 211
- cross-form lookup, 88–91, 257–260
- data entry, 83–84
- data entry on PC, 181
- default value, 122
- deleting, 85
- Exclusive Lookup List, 80

- exporting Lookup Lists, 204–205
- importing Lookup Lists, 207, 209–211
- limiting download using, 172–173
- maximum entries, 83
- maximum entry length, 83
- memory usage, 61–62
- pasting to text fields, 69
- Popup Lists compared, 80
- scripts, 232–233
- sorting, 81
- uses, 84

Lookup List option, 134–135

lookup *value* within *form-name*, 233

lookup *value* within *lookup-list-name*, 232–233

M

machine Data Source, 320, 321, 325–326

math operations

- addition, 224, 230
- assigning to buttons, 252–254
- averaging, 42–43
- cross-record calculations, 230, 253–254
- division, 224, 237–239
- integers, conversion to, 223
- Modulo division, 224
- multiplication, 224
- Quick Sum, 42–43, 45
- rounding, 239–240
- subtraction, 216–217, 224

Max Length Advanced Field Property, 68

Max Length property, 133–134

- .mdb files, 177, 344
 - memory usage
 - checking free memory, 63
 - estimating, 59–62
 - limiting, 62
 - Microsoft Access. *See* Access; linking
 - to external Access databases
 - Microsoft Excel. *See* Excel
 - mid *value start length*, 233
 - Minstrel III wireless modem, 349
 - Model 4000 bar code scanner, 266, 270
 - MS Word reports, 194–196
 - msgbox “*string*”, 233
 - Multi-Mail Pro, 235
 - Multi-Selection List field
 - data entry, 78–79
 - data entry on PC, 181
 - default value, 122
 - multiuser environments
 - accidental overwriting risk, 336
 - licensing, 333
 - Primary Key, 335
 - record sharing privileges, 335–336
 - remote access, 336
 - security, 343–345
 - synchronization options, 337–342
 - synchronization schedule, 336
 - user setup, 334
- N**
- naming forms, 14
 - navigation
 - AutoNavigate, 35–36
 - controlling in form design, 156–160
 - fields, 34–36, 38
 - records, 38
 - view options, 26–27
 - Network HotSync, 339–340
 - Network Setup utility, 337–338
 - New button, disabling, 162–163
 - No Updates access right, 158
 - Novatel Wireless, 349
 - now
 - SQL function, 172
 - variable, 222–223
 - null variable, 223
 - \$number* variable, 222, 226
 - Numeric field
 - data entry, 71
 - data entry on PC, 181
 - default value, 121
 - entering numbers as exponents, 71
 - integers, limiting entry to, 71, 133
 - linked external databases, 296
 - maximum length, 71
 - maximum/minimum value, 132–133
 - number ranges, 71
 - uses, 71–72
- O**
- Open Database Connectivity (ODBC), 317–318. *See also* linking to external ODBC databases
 - open events, 220, 221, 271
 - operators
 - > (arrow) greater than, 170, 227
 - < (arrow) less than, 170, 227
 - >= (arrow, equal sign) greater than or equal to, 227

- <= (arrow, equal sign) less than or equal to, 227
- <> (arrows) not equal, 170, 227
- # (hatch) contains, 224, 227
- % (percent sign) Modulo division, 224
- &t (ampersand) concatenate, 225
- * (asterisk) multiplication, 224
- + (plus sign) addition, 224
- (minus sign) change sign, 223
- (minus sign) subtraction, 216–217, 224
- / (slash) division, 224
- = (equal sign) equal to, 170, 227
- AND, 170
- integer *value*, 223
- length *value*, 223
- OR, 170, 173
- Option 1 of 5 field
 - data entry, 75
 - data entry on PC, 181
 - default value, 122
 - limits, 75
 - uses, 75
- optional *field-number*, 233
- optional from *field-number* to *field-number*, 233
- OR relational operator, 170, 173
- Oracle. *See* linking to external ODBC databases
- ORDER BY, 173–174
- P**
- packet sniffers, 345
- Palm Computing Platform
 - battery life, 3–4, 347
 - clock speed, 4
 - CPU, 4
 - date limitations, 93
 - developer Web site, 354
 - listed, 5
 - naming, 142
 - operating system, 3
- Palm III, 5
- Palm IIIe, 5
- Palm IIIx, 5
- Palm Install Tool, 11
- Palm Query Applications (PQAs), 348
- Palm V, 5
- Palm VII
 - battery life, 347
 - data transmission pricing, 353
 - described, 5
 - Pendragon Forms versus PQA, 348–349
 - sending records to Web site using, 349–353
 - synchronizing options, 347, 349
 - transmit button, 349–353
 - transmitter, 347
 - Web clipping, 347
 - Web site design, 354
- Paradox. *See* linking to external ODBC databases
- parent forms. *See* Single Subform field; Subform List field
- passwords, 320, 326, 330, 343–344
- Pattern option, 135–136
- PC, synchronizing with handheld. *See* synchronization
- pdQ smartphone, 349

- Pendragon Forms Installation dialog
 - box, 11
 - performance issues, 58–62
 - .pff files, 201–202
 - PilotD7A.mdb file, 206
 - Popup List field. *See also* Jump Popup field
 - branching scripts, 248–249
 - cascading, 85–87
 - data entry, 77
 - data entry on PC, 181
 - default value, 122
 - introduced, 16
 - limiting download using, 172–173
 - Lookup Lists compared, 80
 - maximum entries, 79
 - maximum entry length, 79
 - maximum field length, 16, 77
 - maximum item length, 77
 - uses, 78, 80
 - PQAs (Palm Query Applications), 348
 - Primary Key
 - bar codes as, 278
 - behavior during synchronization, 130–131
 - behavior on handheld, 129–130
 - default, 128
 - duplicate Primary Keys, 130–131, 167–168
 - Excel, 130–131, 320
 - frozen forms, 129
 - function, 128
 - linked external Access databases, 129, 286–287, 295–296, 305, 307
 - multiple fields as, 129
 - multiuser environments, 128, 335
 - record sharing, 171
 - selecting, 128–129
 - print {serial|IR}, 234
 - Print Preview mode, 191
 - printing
 - Bachmann Print Manager, 52, 234
 - form designs, 32
 - from handheld, 52–53
 - infrared, 52
 - non-printing fields, 132
 - print buttons, 254–255
 - printer output, 53
 - scripts, 264
 - serial port, 52
 - supported printers, 52
 - using scripts, 234
- Q**
- Qualcomm, 349
 - Qualcomm pdQ Smart Phone, 5
 - Query view, 183–184
 - Quick Sum function, 42–43, 45
- R**
- RAS (Remote Access Server), 339
 - Read-Only attribute, 124
 - Read-Only Text field, 114, 181
 - readonly *field-number*, 234
 - readonly from *field-number* to *field-number*, 234
 - readwrite *field-number*, 234
 - readwrite from *field-number* to *field-number*, 234
 - Record Level Undo, 161–162
 - Record View
 - bar codes, 268

- described, 26–28, 37–40
- hiding Record View button, 158
- parent forms, 108
- setting as default, 160
- subforms, 102
- toggling from Field View, 39–40
- RecordID field, 30, 179, 288
- records
 - Auto-Execute option, 164–165
 - AutoRepeat option, 164
 - blank records, 35
 - changed records, identifying, 46–47
 - creating, 25, 164–165
 - creating on PC, 180–182
 - creation date & time, capturing, 92
 - datasheet display, 29–30, 178–179
 - deleting, 49–50
 - deleting from PC, 182–183
 - deletion, controlling, 165–167
 - deletion, marking for, 114
 - duplicate records, 130–131
 - editing on PC, 180
 - exiting, 26
 - filtering, 44–45, 229
 - incomplete records, 22, 150
 - inserts on handheld, 150, 167–168
 - limiting number on handheld, 62–63, 148–150
 - maximum size, 60
 - modification date & time, capturing, 92, 124
 - multiuser environments, 335–336
 - new records, identifying, 46–47
 - PC display, 178–179
 - printing, 52–53
 - problem records, identifying, 48–49
 - restricting creation to parent form, 162–163
 - retrieving using bar code, 277–278
 - reviewing, 40–42
 - row display, 30
 - selecting, 42
 - sending to Web site using Palm VII, 349–353
 - sharing, 171
 - size estimation, 60–61
 - sorting, 45–46
 - storage limit, 149, 168
 - subform records, 102–106
 - Undo option, 161–162
 - viewing, 26–27
- Records List screen, 41
- reference forms, 88–90
- referencing another form, 88–91
- relational operators, 170
- remote access, 336
- Remote Access Server (RAS server), 339
- Report Generator window, 189
- reports
 - Access, 187–193
 - Excel, 193
 - filtering, 192–193
 - Signature field, 191
 - Word, 194–196
- require *field-number*, 234
- require from *field-number* to *field-number*, 234
- result variable, 222, 225–226, 237
- return statement, 234–235

- reverselookup *value* within *lookup-list-name*, 235
- right *value length*, 235
- rounding numbers, 239–240
- rows, 30
- S**
- sample forms, viewing, 15
- scanner {checkdigit|no_checkdigit}
 - symbology*, 273
- scanner *conversion-code*
 - {enable|disable}, 273
- scanner {enable |disable} *symbology*, 272
- scanner {enable|disable} all, 271
- scanner
 - {system_character|no_preamble} *symbology*, 272
- Screen Preview Report, 32
- Script Checker, 218
- scripts
 - aborting form, 228
 - action statements, 228–236
 - adding to field, 215–216
 - addition, 224, 230
 - Address Book, 230
 - age calculation, 244–245
 - assignment statements, 225–226
 - bar code input control, 270–273
 - beeping, 228
 - branching scripts, 246–250
 - Button fields, 250–257
 - calculate events, 124, 216–217, 219, 220
 - calculations, 236–241
 - calculations across records, 230, 253–254
 - CGI script, 351
 - character strings, matching, 224, 227–228
 - checking syntax, 218
 - click events, 111, 112, 220
 - compiling, 218–219
 - concatenation, 225
 - conditional statements, 226–228
 - counters, 241
 - currency operations, 240
 - date operations, 241–246
 - described, 215–218
 - dialog boxes, displaying, 233
 - distribution, 218–219
 - division, 224, 237–239
 - e-mail, 235
 - enter events, 220, 221, 271
 - exit events, 220, 246, 271
 - exiting, 229, 234–235
 - extracting characters, 232, 235
 - field references, 218
 - font, setting, 229
 - frozen forms, modifying in, 218
 - goto field number, 231
 - hiding fields, 231–232
 - initialize events, 94, 123, 220, 221
 - initializing Date & Time field, 123
 - initializing Date Only field, 123
 - initializing Time field, 123
 - integer conversion, 223
 - invalid values, flagging, 232
 - Jump Popup field branching, 100, 247

- length, returning, 223, 233
- Lookup List branching, 248–249
- lookup operations, 232–233, 235, 257–260
- matching field values, 229
- maximum length, 219
- Modulo division, 224
- multiplication, 224
- open events, 220, 221, 271
- Popup List branching, 248–249
- printing, 264
- printing records, 234
- read-only fields, creating, 234
- read-only fields, overriding, 234
- repeating forms, 261–263
- Required fields, creating, 234
- Required fields, overriding, 233
- result variable, 222, 225–226, 237
- reverse lookup, 235
- rounding numbers, 239–240
- select events, 220, 221–222, 246
- sending records to Web site using Palm VII, 349–353
- subtraction, 216–217, 224
- time operations, 242–244
- unhiding fields, 235
- uniqueness in subform, checking, 229
- user response, testing, 226–228
- validate events, 220, 222, 232, 260–261
- variables, 222–223
- Yes/No Checkbox branching, 250
- Section field
 - data entry, 98
 - hiding Skip Section button, 159
 - Jump Popup field use, 100, 247
 - maximum Section Name length, 98
 - uses, 96–97
- security
 - access rights, 50–51
 - encryption, 344, 345
 - impersonation, 343
 - linked external databases, 329–330
 - multiuser environments, 343–345
 - packet sniffers, 345
 - passwords, 320, 326, 330, 343–344
 - server database storage, 344–345
 - WaveSync synchronization server, 345
- Select Access Database window, 290
- select events, 220, 221–222, 246
- Select Import Source window, 205
- selection lists. *See* Multi-Selection List field
- server database storage, 344–345
- show *field-number*, 235
- show from *field-number* to *field-number*, 235
- Signature field
 - described, 110
 - entering signature, 111–112
 - Legal Terms section, 110, 112
 - reports, 191
 - work orders, 186–187
- signatures. *See* Signature field
- Simple Expense Report form, 15–17
- Single Subform field
 - Autodefault, 125
 - creating, 108
 - data entry, 108–109

Continued

- Single Subform field *continued*
 - hiding subform, 163
 - linking to external database tables, 308–313
 - maximum name length, 109
 - parent/subform correspondence, 108
 - parent/subform separation on PC, 109
 - restricting record creation to parent form, 162–163
 - sorting, restricting, 163–164
 - uses, 107
- Skip Section button, hiding, 159
- sorting
 - Lookup List fields, 81
 - records, 45–46
 - records during download, 173–174
 - restricting on subforms, 163–164
- SPT 1500
 - bar code scanning, 267–268
 - described, 5, 265
 - Diagnostics application, 270
 - filtering records using, 45
- SPT 1700
 - bar code scanning, 5, 265
 - described, 5
 - Diagnostics application, 270
- SPT 1740, 5
- SQL. *See* Structured Query Language (SQL)
- Start(Programs(Pendragon Forms 3.0(Install Forms 3.0 on Handheld, 11
- Start(Programs(Pendragon Forms 3.0(Pendragon Forms Manager, 11
- Start(Programs(Palm Desktop(Palm Desktop, 142
- starting Pendragon Forms
 - handheld, 33
 - PC, 11
- statements
 - action statements, 228–236
 - assignment statements, 225–226
 - conditional statements, 226–228
 - defined, 225
- storage limit, 149, 168
- Structured Query Language (SQL)
 - introduced, 170
 - now function, 172
 - ORDER BY clause, 173–174
 - WHERE clause, 170
- Subform field. *See* Subform List field
- Subform List field
 - Autodefault, 125
 - copying fields from parent to subform, 102, 103
 - creating records, 102–104, 105
 - data persistence, 107
 - deleting records, 107
 - display key, 102, 104–105, 106
 - existing subform records, 104–106
 - hiding subform, 163
 - linking to external database tables, 308–313
 - maximum subform name length, 104

- multiple subforms, 106
- nesting subforms, 106
- parent/subform linking, 101–102
- parent/subform relationship, 100, 105
- parent/subform separation on PC, 107
- read-only subform fields, 103
- restricting record creation to parent form, 162–163
- sorting, restricting on subform, 163–164
- subforms. *See* Single Subform field; Subform List field
- Symbol Technologies, 265
- synchronization
 - Archive Date, 149, 169
 - bi-directional synchronization, 7, 30
 - changing existing records, 30
 - data persistence options, 21–22, 148–149
 - data transfer rate, 58–59
 - Do Not Upload to PC field property, 131–132
 - download, limiting by date, 172
 - download, limiting by Selection List criteria, 172–173
 - download criteria, default, 150, 171
 - duplicate records, 130–131, 167–168
 - form designs, 24–25, 146, 150–151, 202
 - Forms conduit, 6–8, 137, 138–139
 - frequency, 63–64
 - HotSync Manager described, 4, 138
 - incomplete record options, 22, 150
 - inserts on handheld, 150, 167–168
 - linked external databases, 298–301, 307, 310–313, 315
 - modem HotSync transfers, 64
 - multiuser environments, 336, 337–342
 - network HotSync transfers, 64–65
 - optimizing, 66
 - overwriting existing PC records, 150, 167–168
 - Palm VII, 347, 349
 - Primary Key behavior, 130–131, 167–168, 171
 - problem records, 48–49
 - process described, 139–140
 - record order after HotSync, 45
 - remote access, 336
 - sorting during download, 173–174
 - speed, 58–59, 64–66
 - SQL involvement, 170
 - storage limit, 149, 168
 - uploading to PC, 28–30, 46–49
 - user groups, 24, 146
 - user management, 140–146
 - WaveSync synchronization server, 66, 341–342
 - wireless synchronizing, 347
 - work orders, 185–186
- system requirements, 12

T

- Table Selector window, 291
- text fields. *See* Freeform Text field;
 - Read-Only Text field
- Time Checkbox field
 - data entry, 96
 - data entry on PC, 182
 - uses, 96
- time display, 95
- Time field
 - 24 hour clock, 95
 - data entry, 95
 - data entry on PC, 182
 - default value, 95, 122, 123
 - initializing using scripts, 123
 - uses, 95
- time scripts, 242–244
- time-to-live options. *See* data
 - persistence
- TimeStamp field
 - Archive Date, 169
 - data persistence, 169
 - default Primary Key, 171
 - introduced, 30
 - linked external database issues,
 - 288
 - modification date, 123–124
- Toggle Change Flags, 47
- Toggle Default View, 40, 160
- Tools(Users, 142
- transmit multimail “*e-mail address*”,
 - 235, 256–257
- transmit palmnet “*address*”, 236,
 - 349–353

U

- UnitID field, 30, 179, 288
- uploading to PC. *See* synchronization
- user groups. *See also* user
 - management
 - All Users Group, 144
 - creating, 144–145
 - default group, 143–144
 - deleting forms, 145
 - deleting users, 145
 - described, 143
 - form design synchronization, role
 - in, 24, 146
 - importing, 207
 - multiuser environments, 334
 - user management. *See also* user
 - groups
 - active users, 140
 - adding users, 140–141
 - deleting users, 142
 - handheld user name, 142, 171
 - multiuser environments, 334
 - passwords, 320, 326, 330, 343–344
 - user lists, 140–142
 - user lists, importing, 207
 - user name, displaying, 11
 - user response, testing, 226–228
 - ##USERNAME##, 171, 300–301
 - [UserName] =, 171
 - UserName field, 30, 169–170, 179,
 - 288–289

V

- validate events, 220, 222, 232,
 - 260–261

variables, 222–223
version information
 CD-ROM, 10
 Pendragon Forms Version 2, 206

W

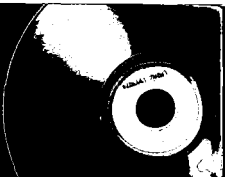
WaveSync synchronization server, 66,
 341–342, 345
WaveWare Communications, 341
Web clipping, 347
where (SQL function), 170
Word reports, 194–196
work orders
 Completion Checkbox, 185
 introduced, 184–185
 sending to handheld, 185–186
 signatures, 186–187

World Wide Web
 Palm developer site, 354
 sending records to, 349–353
 site design for Palm VII, 354

Y

Yes or No Checkbox field. *See also*
 Completion Checkbox field
 branching scripts, 250
 data entry, 74
 data entry on PC, 182
 default value, 122
 limiting download using, 172–173
 limits, 74
 uses, 74

THE OFFICIAL PENDRAGON FORMS™ FOR PALM OS®



CD-ROM includes
Pendragon Forms 3.0
evaluation version!
PLUS evaluation versions of Bachmann
Print Manager, WaveSync synchronization
server, MultiMail PRO, and all sample
forms from the book.

Pendragon Forms software lets you transform a Palm OS organizer into a powerful mobile data-collection device. This complete guide, written by two senior executives at Pendragon Software, gives you all the inside tips and tricks you need to create robust forms and scripts that are tailored to your specific needs — from expense reports and work orders to merchandise inventories and data from the field. You'll discover how to automate with bar code readers, link Pendragon forms to external databases, and more — all with no programming skill required.

Working Solutions for Data-Collection Challenges:

- Create custom forms for fast and efficient data collection
- Maximize accuracy and productivity with proven form-design techniques
- Write powerful scripts that perform calculations, branching, and more
- Master synchronization rules for controlling data flow between Palm OS devices and PCs
- Link Pendragon Forms to external Microsoft Access, SQL, and Oracle databases
- Find out how to print from Palm OS devices
- Set up Pendragon Forms for a multi-user environment

About the Authors

Debra Sancho is a vice president at Pendragon Software and the author of much of the technical documentation for the company's products.

Ivan Phillips is the president of Pendragon Software and the lead developer for Pendragon Forms. He has been developing software for more than 15 years and is the coauthor of two books on Java.

Register to win!

my2cents.idgbooks.com

www.idgbooks.com

Reader Level: Beginning to Advanced

Shelving Category: Palm/Database

System Requirements: PC: Access 97 or Access 2000 recommended; Palm Desktop 3.0; Windows 95/98, or Windows NT4/SP3. **Handheld:** Palm OS 3.0 or later.

\$39.99 USA

\$59.99 CANADA

£36.99 UK incl. VAT



Pendragon Forms is a trademark of Pendragon Software Corporation. Palm OS is a registered trademark of Palm Computing, Inc., 3Com Corporation, or its subsidiaries. The IDG Books Worldwide logo is a registered trademark under exclusive license to IDG Books Worldwide, Inc., from International Data Group, Inc. M&T Books and the M&T Books logo are trademarks of IDG Books Worldwide, Inc.

IDG Books Worldwide
An International Data Group Company
Foster City, CA 94404

Printed in the USA

RPX-1012

Page 444 of 444

ISBN 0-7645-4651-1

