
SmartList To Go[®] 3.0

HANDHELD REFERENCE GUIDE

CREATE, MANAGE, AND SHARE INFORMATION ON YOUR HANDHELD

DataViz, Inc.

MERRITT CORPORATE WOODS • 612 WHEELERS
FARMS ROAD • MILFORD, CT 06460

PHONE 203-874-0085 • www.dataviz.com

© Copyright DataViz, Inc, 2004

Table of Contents

1. INTRODUCING SMARTLIST TO GO 3.0	4
1.1 Using this Manual	5
1.1.1 The Scope of this manual	5
1.1.2 How to Use this Manual	5
1.1.3 Conventions	6
1.2 Contacting Sales and Technical Support	7
1.2.1 Required Information to Have Before Calling	7
1.2.3 Contact Information	8
2. GETTING STARTED WITH SMARTLIST 3.0	9
2. GETTING STARTED WITH SMARTLIST 3.0	9
2.1 Handheld Compatibility and System Requirements	9
2.2 Installing the SmartList To Go Handheld Application	10
3. AN OVERVIEW OF SMARTLIST TO GO	11
3.1 Launching the SmartList To Go Handheld Application	12
3.2 SmartLists	13
3.3 A Tour of MyBooks	14
3.3.1 Opening a SmartList	15
3.3.2 The List View Screen	15
3.3.3 Records	18
3.3.4 Closing a SmartList	21
4. DESIGNING SMARTLISTS.....	23
4.1 SmartList Creation.....	24
4.1.1 Creating a SmartList	24
4.1.2 Setting SmartList Properties	25
4.1.3 Using the SmartList Main Menu	28
4.1.4 Renaming a SmartList	31
4.1.5 Deleting a SmartList.....	31
4.2 Designing a SmartList	32
4.2.1 The Field Editor Screen	34
4.2.2 The Form Designer Screen.....	38
4.2.3 The View Editor Screen	44
4.2.4 The List Editor Screen	47
4.3 Records	51
4.3.1 Creating Records	51

4.3.2 Duplicating Records	51
4.3.3 Deleting Records.....	52
4.3.4 Deleting All Records	52
5. USING ADVANCED FUNCTIONS	53
5.1 Categories	54
5.1.1 Creating Categories	55
5.1.2 Assigning Records to a Category	56
5.1.3 Displaying by Category	57
5.2 The Expression Field	58
5.2.1 Creating an Expression Field.....	59
5.2.2 The Expression Builder Screen	59
5.3 The Conditional Expression Field	63
5.3.1 The Parts of a Conditional Expression	63
5.3.2 Creating a Conditional Expression.....	64
5.3.3 The Conditional Expression Properties Screen	65
5.4 The Advanced Expression Field	68
5.4.1 Creating an Advanced Expression Field.....	69
5.4.2 The Advanced Expression Builder Screen	69
5.5 Beaming	80
5.5.1 Beaming a SmartList.....	81
5.5.2 Beaming a Record	82
5.6 Cloning a SmartList	84
5.7 Filters	85
5.7.1 Configuring Filters	85
5.7.2 Applying Filters.....	90
5.8 Filtered Operations	91
5.9 Security	94
5.9.1 Setting a Password	95
5.9.2 Changing a Password.....	96
5.9.3 Removing a Password	98
5.10 The Image Field.....	99
5.10.1 Creating and Configuring an Image Field	99
5.10.2 Using the Image Screen	100
5.10.3 Adding an Image Field to the Form.....	103
6. RELATING SMARTLISTS	104
6.1 Relating SmartLists using DB Join and DB Lookup Fields	105
6.1.1 Why use DB Join and DB Lookup Fields?	105
6.1.2 DB Join.....	107

6.1.3 DB Lookup	109
6.2 Relating SmartLists to the Palm Address Book.....	110
6.2.1 Address Join	111
6.2.2 Address Lookup	112
6.3 Relating SmartLists using One-to-Many	114
6.3.1 The One-to-Many Field	114
6.3.2 Enhancing a One-to-Many Relationship with DB Join and Key Fields	118
APPENDIX A. SMARTLIST TO GO FIELD TYPES.....	119
A.1 Supported Field Types	119
A.2 Field Type Capabilities	122
APPENDIX B. SMARTLIST TO GO SCREEN AND CARD SUPPORT	123
B.1 Palm OS Screen Support.....	124
B.2 Palm OS Expansion Card support.....	125
APPENDIX C. SMARTLIST TO GO PLUG-IN SUPPORT.....	126
Date Book Plug-In	126
Timer Plug-In.....	127

1. Introducing SmartList To Go 3.0

SmartList To Go 3.0 is a relational database management application designed for use on the Palm or other handhelds using the PalmOS.

- **1.1 Using this Manual**
 - **1.1.1 The Scope of this Manual**
 - **1.1.2 How to Use this Manual**
 - **1.1.3 Conventions**
 - **1.1.4 Documentation for Our Other Products**
- **1.2 Contacting Customer Service and Technical Support**
 - **1.2.1 Online Resources**
 - **1.2.2 Required Information to Have Before You Call**
 - **1.2.3 Contact Information**

1.1 Using this Manual

1.1.1 The Scope of this manual

This manual describes how to use the SmartList To Go handheld application. Included in this manual are complete instructions and examples describing how to:

- download and install the application,
- create databases on your handheld and configure them according to your project needs,
- sort records according to field headings, categories or custom designed filters,
- relate databases to each other using Joins and One-to-Many relationships,
- print or export databases or individual records,
- acquire assistance with any technical difficulties you might encounter.

While the functionality described in this manual is current as of its release, it may not reflect new functionality or bug corrections implemented after the manual's date of publication. For the most complete and current information, please refer to the release information on our Web site at www.dataviz.com.

Please be aware that the scope of this manual encompasses the SmartList To Go handheld application only. This manual must and does assume that users are already versed in the use of their handheld device. For information regarding the use of your handheld device, please refer to the manufacturer's documentation for that device.

1.1.2 How to Use this Manual

The purpose of this manual is to serve users as both a set of complete instructions and a quick reference guide. By that we mean that this manual is designed to be used in two distinct ways. First, feel free to read this manual from front to back like a book. Doing so will guide you through the process of learning how to use the SmartList To Go handheld application. That process begins with learning your way around SmartList To Go handheld databases, proceeds to the basics of creating your own databases, and finally addresses the advanced functions that allow you to design numerical and conditional expressions and to interrelate individual databases with each other or Palm applications like Memo Pad and Address Book.

The second method of using this book is to provide a quick reference for those who, for the most part, already know the SmartList To Go handheld application, or are comfortable enough with databases in general to prefer working things out largely for themselves. To accommodate this method, each section in this manual is designed to lead off with quick and detailed information describing the functionality of the interfaces that section addresses. Using the Table of Contents at the front of the manual, users can quickly find the sections or pages with the answers they are looking for. The design of the pages will make the specific items of information they are looking for easy to find.

1.1.3 Conventions

Throughout this manual, we will use certain visual conventions to set off different kinds of information. The purpose of this is to make it easier for you to understand what you are reading. Among these conventions are:

- The names of screens, commands, icons and other distinct items in the SmartList To Go interface are capitalized, as shown below:

To add this new field to the form, open the Field Designer screen.

- Any text depicting what you will see on the handheld screen will be displayed in a sans serif font that imitates the screen display. Further, such text will be presented in separate paragraphs that are indented to further differentiate it from other text. For example, when an instruction directs you to enter a value for the text field "Last Name", you would see the following:

Last Name: Jones

- Instructions in this manual at times refer to items of information generally so that the word or phrase in the manual represents the actual text that applies to your specific information. These variables, as we will refer to them, appear in a bold-face font to differentiate them from text that may be taken literally, as shown below:

Enter Your Last Name: **Your Last Name**

-OR- Enter **Today's Date** in the input box provided.

- Instructions that guide you in selecting items in a menu path will separate items with the ">" symbol. Also, such menu paths will appear in bold-face, as in the example below:

Select **Tool > Design Views** to open the View Designer screen.

- Shortcut commands allow you to accomplish common tasks quickly by bypassing the normal menu commands. As in most Palm OS applications, you use such commands by making a forward slash with the stylus - starting from the lower left corner of the graffiti pad, draw the stylus across to the upper right corner - followed by the shortcut's identifying letter. For example, the shortcut for creating a new database is a forward slash followed by the letter Y. This manual will refer to shortcuts by typing the forward slash and the letter together in bold face, as shown below:

To create a new database, open the **Main Menu** and select **SmartList > Create New SmartList** or use the shortcut **/Y**.

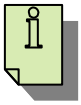
In addition to these conventions, this manual includes the following kinds of additional information that are set off by their icons and borders.



Note: Notes provide extra items of information that expand upon the nearby text. For example, a note might direct you to another location within the manual or elsewhere that provides additional information about the current topic.



Warning: Warnings set off cautions and other items of information that will help you to avoid situations that might result in a loss of data or temporary or permanent damage to your software or damage to your hardware. Always take a moment to stop and read these cautions, as the information they contain will always be valuable to you.



Tip: Tips inform you of key items to remember as you proceed or of tricks you can use or of work-arounds you might employ if they are useful to you. Anything that might enable you to streamline your processes and increase efficiency will be included in a tip.



Example: Examples take the concepts discussed in the section and show you how they might be applied to better your understanding of how a function or capability might be used to satisfy your project needs.

1.2 Contacting Sales and Technical Support

In This Section . . .

- 1.2.1 Required Information to Have Before Calling
- 1.2.2 Contact Information

1.2.1 Required Information to Have Before Calling

In order for our Sales and Technical Support staff to serve you better, please have the following items available and close at hand before you call. Some items may depend on your specific issue, but items marked with an asterisk next to them are always required for all calls.

- The name and email address under which you registered your copy of SmartList To Go .
- The registration code provided to you when you purchased SmartList To Go.
- The handheld with SmartList To Go installed must be with you as our support personnel may ask you to perform certain tasks to better determine the nature of the problem or to correct the problem.
- If you are calling in regards to a software problem, please be prepared to describe the errant

behavior fully, including, if applicable, taking exact notes of any error messages you may be receiving and what causes them to appear.

- If you are calling in regards to a problem regarding synchronization with a desktop application, make sure you are close to your desktop as our support personal may ask you to perform certain tasks to better determine the nature of the problem or to correct the problem.

1.2.3 Contact Information

General:

Phone: 203-874-0085

Fax: 203-874-4345

DataViz, Inc.

Merritt Corporate Woods

612 Wheelers Farms Road

Milford, CT 06460 USA

Sales & Upgrades:

Phone: 800-733-0030 or 203-874-0085

Fax: 203-874-4345

Our Sales department is available Monday - Friday 9 am - 5 pm EST. Please keep in mind that our offices are not open on weekends and [U.S. holidays](#).

Tech Support:

DataViz offers a number of options for tech support. To learn more about each option, please visit [Support Options](#).

Web Site:

www.dataviz.com

2. Getting Started with SmartList 3.0

In This Chapter . . .

- 2.1 Handheld Compatibility and System Requirements
- 2.2 Installing SmartList

2.1 Handheld Compatibility and System Requirements

- Mac OS 10.1 or greater
- Palm OS 3.5 or greater
- HotSync 3.1 or greater
- 750kB free space on the handheld

SmartList To Go 3.0 for PalmOS is compatible with any handheld or other device that operates using the PalmOS® version 3.5 or higher, including but not limited to:

Garmin

iQue 3600

Handera

330

Handspring

Treo 180, Treo 180G, Treo 270, Treo 300, Treo 600, Treo 90, Visor Edge, Visor Neo, Visor Platinum, Visor Prism, Visor Pro

Kyocera

7135 Smartphone, QCP-6035 Smartphone

Palm

i705, IIIc, IIIxe, m105, m125, m130, m500, m505, m515, Tungsten C, Tungsten E, Tungsten T, Tungsten T2, Tungsten T3, Tungsten W, VIIx, Zire 21, Zire 71

Samsung

i300, i330, i500

Sony

PEG-N610C, PEG-N710C, PEG-N760C, PEG-NR70, PEG-NR70v, PEG-NX60, PEG-NX70v, PEG-NX80V, PEG-NZ90, PEG-S300, PEG-S320, PEG-S360, PEG-SJ20, PEG-SJ22, PEG-SJ30, PEG-SJ33, PEG-SJL10, PEG-T415, PEG-T615C, PEG-T665C, PEG-TG50, PEG-TJ25, PEG-TJ35, PEG-UX40, PEG-UX50

For additional information on whether your handheld or other device currently uses or can be

upgraded to use PalmOS® 3.5, please refer to the device documentation provided by the manufacturer.

To upgrade your device to PalmOS® 3.5, visit the Palm Web site at www.palmone.com.

2.2 Installing the SmartList To Go Handheld Application

1. The download you obtained is a cd image that

installation program will start automatically when you insert the SmartList To Go CD into your desktop CD-ROM drive, or open Microsoft Explorer, select the CD-ROM drive and doubleclick the file **setup.exe**.

2. Follow the InstallShield Wizard instructions to install to your desktop.
3. HotSync to install the application and featured SmartLists to your handheld device.

The installation process is now complete.

3. An Overview of SmartList To Go

The SmartList To Go handheld application is a powerful relational database manager. It is conceptually and functionally similar to some of the desktop database applications you may already be familiar with, such as Microsoft Access, but with a straight forward design and structure for optimal performance in the handheld environment.

The purpose of this chapter is to introduce you to SmartList To Go. The following sections will take you on a tour through existing databases, explaining new vocabulary and illustrating the broad concepts of what SmartList To Go is and what it can do. This will provide you with a strong foundation for understanding future chapters that will explain how to create and use databases.

In This Chapter . . .

- **3.1 Launching the SmartList To Go Handheld Application**
- **3.2 SmartLists**
- **3.3 A Tour of MyBooks**
 - **3.3.1 Opening a SmartList**
 - **3.3.2 The List View Screen**
 - **3.3.3 Records**
 - **3.3.4 Closing a SmartList**

3.1 Launching the SmartList To Go Handheld Application

When you install SmartList To Go on your handheld device, the installation added the SmartList To Go icon to your Applications Launcher screen. To initiate SmartList To Go, select this icon. SmartList To Go will start up and display the SmartList To Go Home screen. The SmartList To Go Home screen lists the current SmartLists.



SmartList To Go				▼ All
SmartList	Recs	Mod		
▼ Calorie Chart	309	2/9		
▼ Calorie Tracker	2	2/9		
▼ CDCollection	5	2/9		
▼ DVD Collection 1.	1	2/9		
▼ Exercise Assistan	33	2/9		
▼ Medical Records	4	2/9		
▼ My Books	11	2/9		
▼ Personal Trainer	3	2/9		
▼ Prescription	2	2/9		
▼ Start Here	11	2/9		

The SmartList To Go Home screen

If SmartList To Go opens on a screen other than the Home screen, click on the **Home** icon in the lower left corner of your screen.



3.2 SmartLists

In SmartList To Go, a **SmartList** is a database. Simply stated, it is a collection of one or more individual records contained in a single file. Each record stores the same types of information types as every other record in the SmartList, but each record describes a separate individual item in a group.

For example, a SmartList named "MyBooks" may include book title, author's name and other data for a collection of books. Each record would provide information describing a different individual book, but every record would provide the same kinds of information about each book.

Definition: A SmartList is a single file containing one or more records, each record providing the same types of information describing items in a group. A SmartList may include up to 65,000 individual records.

3.3 A Tour of MyBooks

With SmartList To Go you have the option to install Sample SmartLists from the SmartList desktop application. Throughout this manual we will refer to some of these sample SmartLists as examples of the concepts we describe. In this section, we will take you on a tour through one of these SmartLists to introduce the new terminology you will use throughout the rest of this manual.

The name of this sample SmartList is MyBooks. This SmartList describes a personal collection of books. If you have the sample named MyBooks, you may want to open it now and follow along.

In This Section . . .

- **3.3.1 Opening a SmartList**
- **3.3.2 The List View Screen**
 - **3.3.2.1 Sorting Records in a List View**
 - **Selecting a List View**
 - **Changing the Column Width**
 - **Sorting by Column**
 - **Filtering the List**
 - **Displaying by Category**
 - **3.3.2.2 The List View Toolbar**
- **3.3.3 Records**
 - **3.3.3.1 Opening Records**
 - **3.3.3.2 The Record Screen**
 - **Fields**
 - **Values**
 - **Field Types**
 - **Forms**
 - **Labels**
 - **Static Text**
 - **3.3.3.3 Closing Records**
- **3.3.4 Closing a SmartList**

3.3.1 Opening a SmartList

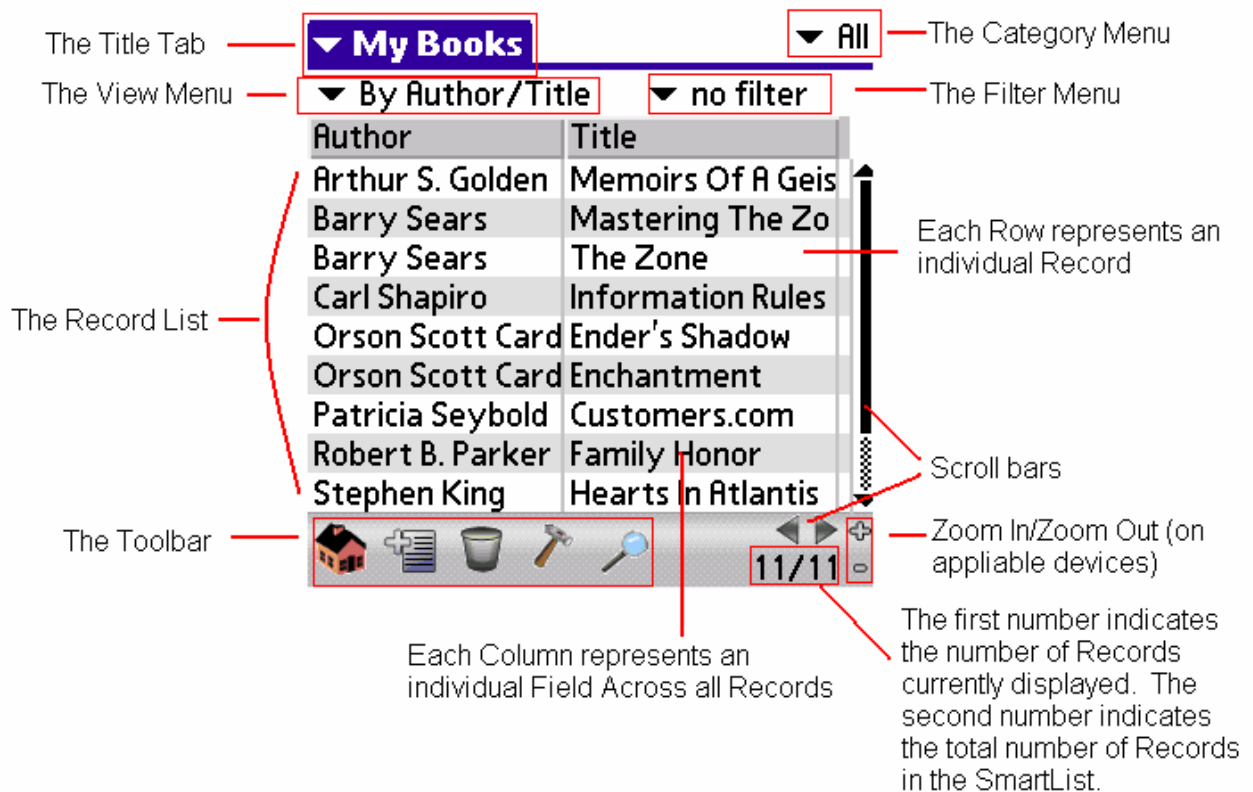
To open a SmartList from the Home screen, click on the name of the SmartList. This will open the file and display the List View screen.

We'll discuss the List View screen in more detail in a moment, but for now, just know that it provides a quick, limited view of all the records in the SmartList at a glance. Notice that each record represents data describing a different book, but every record provides the same information: the book's title and the author's name.

3.3.2 The List View Screen

MyBooks will open to display the **List View screen**. This screen provides a list of all the records in MyBooks and displays some of the data contained in each record, in this case the book title and the author.

Take a moment to notice some of the key elements of the List View screen.



At the top of the screen, the Name tab displays the name of the SmartList. This tab is visible from any screen in the SmartList, and clicking on this tab will open the main menu. In addition, the Recently Used SmartList pull-down menu to the left of the name on the Name tab contains a list of the 5 other SmartLists you opened most recently. Use this menu to quickly move from one commonly used SmartList to another.

Dominating the center of the screen is the List View itself, providing a list of all the records in the SmartList in tabular form. In this Table, each column represents a field and each row represents a

unique record. The column headers indicate the name of the field. When you create your own SmartList, you can control the appearance of this list, including which items are listed and what kinds of information are visible by using the View Select, Filter Select and Category Select menus. To the right of the list is a scroll bar that allows you to scroll your view of the list up or down.

Just below the tab you will find two pull-down menus and a third menu is in the upper right corner of the screen. The menu on the left is the View Select menu that allows you to view the records in a SmartList in a variety of different list configurations that you design. For example, the default view for MyBooks lists the books in alphabetical order by title and shows the book title and author's name. Another view, Pages, displays the title of the book and the number of pages left to read, ordering the list by page numbers, not title.

The menu on the right is the Filter Select menu that allows you to display limited groups of the total records based on the data within them. For example, select the Borrowed filter to list only those books that have been lent out to others. Select no filter to display all the books in the SmartList.

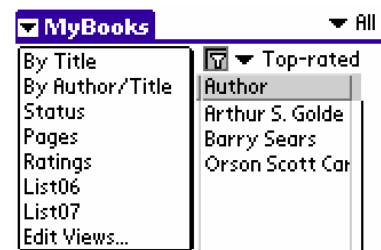
The third pull-down menu is the Category Select menu. Use this menu to display records by category. For example, select **Fiction** to select only those books that have been categorized as fiction. Select **All** to return to displaying all the records.

3.3.2.1 Sorting Records on the List View Screen

The List View screen offers four different methods of sorting and filtering records to suit different needs. These methods include selecting a view, sorting by column, filtering the list, and displaying by category.

Selecting a View

Just below the tab on the List View screen is the View Select menu. The **View Select** menu is a pull down type menu that allows you to select from a number of customizable views. A **View** is a tabular display of the records in a SmartList which displays only selected types of information about the records. You may design up to 12 different views to display different combinations of information to accommodate a variety of situations.



For example, the default view is called "By Title," and it displays the books' titles and authors' names only, providing a complete list of all the books in the SmartList.

Click on the **View Select menu** and select **Status** to display the Status view. The Status view, instead of showing all the books in the collection, instead shows only those books that are currently lent out. This view shows that information that the book is lent, what the title of the book is, and who has the book in order to let you know where all your books are at a glance.

Click on the **View Select menu** again and select **Pages** to display the Pages view. The Pages view provides a quick list of the books you are currently reading and how far along you are.

You'll notice that both the Status and Pages views do not list every book on the list. In the Status view, books that are not lent out do not appear. Similarly, the Pages view does not display books you are not currently reading. This is accomplished by combining the view design with a filter. We will discuss how to design views and how views and other methods of sorting can work

together in Chapter 4, but for now we just want you to understand how views can work.

Before continuing to the next section, click on the **View Select menu** and select **By Title** to return to the complete list of records.

Changing the Column Width

You can adjust the width of columns in a List View to better display data and labels using two methods.

First, you may place the stylus on the border between column labels and drag left or right to change the width of the column to the left. The columns to the right will shift to accommodate the change while maintaining their current width.

You can also adjust the width of a column to automatically accommodate the width of the column label. To do this, click on the column label. This will open a pop-up menu with three options in it. Two are arrows for sorting records as described in the next section. Select the third option, two inward pointing arrows that represent Autosize. The width of the column will automatically resize to fit the width of the column label.



If you alter the width of a column so that content in one or more records is wider than the available space, SmartList To Go will not display truncated information as this could be misleading when viewed at a glance. Instead, the SmartList To Go handheld application will replace the data display with a series of pound signs (#). To view the complete data, adjust the width of the column to accommodate the full content in the View screen or select the record to view the full content in the Record form.

Sorting by Column

The labels at the top of each column indicate what item of information appears in that column. You can sort the items in your view according to the information in any column.

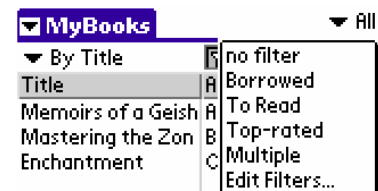
To sort by any column, click on the column label. An arrow menu will open as shown in the figure to the right. Select the down arrow to list records according to information in that column in increasing alpha-numerical order. Click the up arrow to list records in reverse alpha-numerical order.

Filtering the List

To the right of the View Select menu is the Filter Select Menu. A **filter** enables you to display a limited list of records by showing only those records that contain certain information in certain columns.

You should now see that "no filter" is currently selected. Click on the **Filter Select menu** and click on **Borrowed** to see the Borrowed filtered list. The Borrowed filter shows only those records that indicate "Borrowed" in the Status column; this selection filters out all records with any other value.

Click on the **Filter Select menu** again and select **Top Rated** to the Top Rated filtered list. This list shows only those books with the highest number of stars under rating. The filter can do this even though the rating column is not displayed.

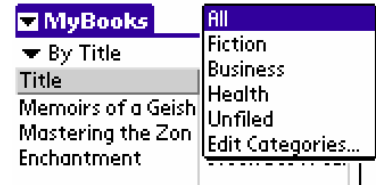


You can configure and use up to 12 different filters. Each filter and test records according to up to three different criteria. You can even select more than one filter at a time to display only those records that are common to both filters.

Before you continue to the next section, click on the **Filter Select menu** again and select **no filter**.

Displaying by Category

Located in the upper right hand corner of your screen, you will find the Category Select menu. A category is yet another way of classifying records. The Category Select menu enables you to list records from only one category at a time.



Click on the **Category Select menu** and select **Fiction** to see a list of only those books in the fiction category. Click again and select **Business** to see only those books in the Business category.

Before continuing to the next section, click on the **Category Select menu** and select **All** to display books in all categories once again.

3.3.2.2 The List View Toolbar

Along the bottom of the screen, the five buttons you see displayed make up the SmartList To Go Tool bar. These buttons allow you to quickly return to the Home screen, enter new records, delete records, access design screens, and search for keywords.

The SmartList To Go Tool bar



Home



New Record



Delete Record



Tools



Search

Finally, in the lower right corner of the screen, the two numbers you see indicate, in order, the number of records currently displayed in the List View screen and the total number of records in the SmartList.

3.3.3 Records

3.3.3.1 Opening Records

To open a record from the List View screen, click anywhere on the row that contains the record. This will open the record and display the Record screen.

3.3.3.2 The Record Screen

From the List View screen, select the record for Stephen King's book *Hearts in Atlantis* to open the Record screen. The purpose of the **Record screen** is both to display all the information contained within a record and to provide an interface for changing that information.

A **record** is a collection of fields (variables) and their values. Taken together, the fields and their values describe one item in a group. Each record contains the same kinds of information, or fields, but the values entered describe just one item.

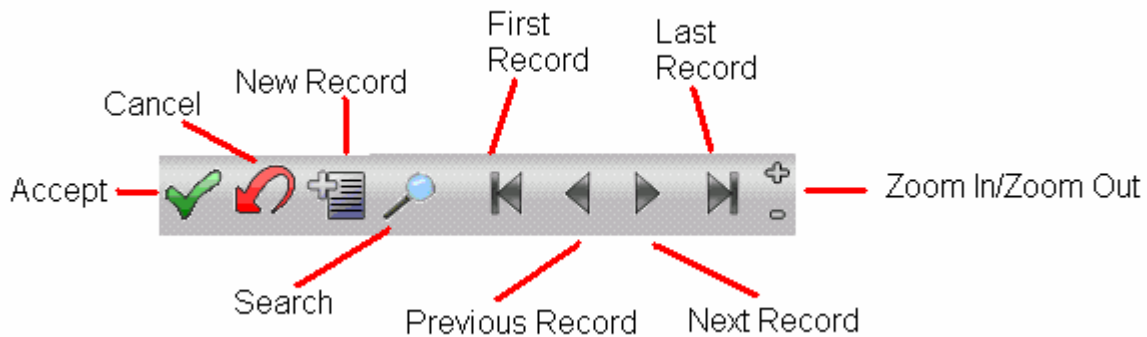
Definition: A **record** is a group of one or more fields and their corresponding values that describe one item in the group represented by the SmartList. A SmartList may include up to 65,000 records. A record may contain up to 80 fields.

For example, in MyBooks, each record contains information describing each book's title, author, publisher, number of pages, ISBN and more.

Take a few moments to take notice of some of the Record screen's key elements.

The tab in the upper left corner of the screen now displays the record's number and how many records are in the SmartList. Just as in the View List screen, you can click on this screen to open the main menu.

In the lower right hand corner of the screen is the **Record Tool bar**. Use this tool bar to move, display the previous or next record, the first or last record, create a new record or search for a record.



Dominating the center of the Record screen is the record's Form. The Form contains the fields with their labels and spaces to enter values. These elements work closely together to control what data is entered, to store the data and to display it on the screen. Understanding these elements is key to understanding the SmartList To Go handheld application.

Fields

In the SmartList To Go handheld application, a **field** is a named placeholder for storing a certain kind of information you can then recall for review or manipulation by referring to the field's name. For example, in the SmartList titled MyBooks, there is a field called "Title" that stores the book's title, "Author" that stores the book author's name, "ISBN" that stores the book's ISBN number, and many more.

Definition: A **field** is a named placeholder for storing data in SmartList To Go. A field's **type** controls how you can enter, display and manipulate data. There are currently 22 field types to choose from, and a record may contain up to 80 individual fields.

Values

A **value** is the actual data that you store in a field. For example, in the sample record above, we can see that the name "Stephen King" appears in the field labeled "Author." "Stephen King" is the field's value. Taken together, fields and their values make up a record the

Definition: A **value** is the information stored within a field.

same way records make up a SmartList.

Field Types

The SmartList To Go handheld application accommodates a number of different kinds of fields. You determine what **field type** to use based upon what kind of data it will store because it controls how the values can be manipulated. For example, you will store the author's name in the field called "Author," so you will select the **text** field type. In the field called "Paid," however, you will enter a decimal number value that represents how much you paid for the book, so you will select the **float** field type to enable the entry of numerical values with a floating decimal point.

You could also select the **text** type for the "Paid" field, but this is a good example of how field type selection controls how the SmartList To Go handheld app can manipulate values. If you select to enter the amount paid as text, then SmartList To Go will consider it to be literal text and nothing more. If you select the float field type, the SmartList To Go handheld app will consider the value to be numerical and will, therefore, allow you to perform mathematical calculations using the value in this field. For example, you could create a second **float** type field called "Retail Value" and a third field called "You Saved". You could then configure the fields to accept the amount you paid in the "Paid" field, to accept the current retail value of the book in the "Retail Value" field, to automatically calculate the difference between these two values and to display the result in the "You Saved" field.

Currently, the SmartList To Go handheld application accommodates the following field types:

- Text
- Memo
- Integer
- Long Integer
- Floating Point
- Date
- Time
- List
- Checkbox
- Radio Buttons
- Expression
- Conditional Expression
- Advanced Expression
- Key
- Auto Incremental
- DB Join
- DB Lookup
- Address Join
- Address Lookup
- One to Many
- Image
- SmartList To Go Plug-In

We will discuss Field types in more detail later in this manual.

Forms

SmartList To Go's data entry **forms** and field labels provide a customizable, user friendly interface for both creating and reviewing records.

Definition: A **Form** is a screen in the SmartList To Go handheld application that provides a user-friendly interface for entering and reviewing record data.

Definition: A **field type**

indicates what kind of information will be entered in a field and how it is to be used.

Recd 5 of 12 ▼ Fiction

Title: Hearts in Atlantis
Author: Stephen King

General | Location | Reading

ISBN: 684853515
Publ: Scribner
Pages: 512
Edition: ▼ Paperback
Release: 9/14/99
Rating: ▼ ****
Paid: 20.00

As we discuss forms and labels, take another look at the sample Record screen from MyBooks. Remember that a field is really just a named place holder for data. Once you have created your fields, you'll need to place those fields on a form as in the form to the right.

Each field in a form occupies one entire line, but you can manage many fields by dividing multiple tabs. The example to the right shows three tabs, "General," "Location" and "Reading." Each of these tabs displays a different set of fields. The fields above the tabs remain visible no matter what tab you select.

Only add those fields you want to display on the Record Screen. For example, you may use a Key field to ensure all records are unique, but because the user cannot manipulate Key values, you may not wish to include the Key field on the form.

Labels

Look at the sample record for Stephen King's book *Hearts in Atlantis* above. There are many fields on this form, and each field is intended to accept a different item of information. You know which information to enter in each field, however, because the field labels guide you. Field **labels** increase and enhance the usability and user-friendliness of a form by indicating what information each field is intended to accept.

Definition: A Label is text that accompanies each field input on a form. The purpose of this text is to indicate to users what kind of data to enter in each field. Labels are often identical to or closely based on the field's name, but this is in no way a requirement; designers are free to label fields using whatever text seems most appropriate.

It is important to note that the **label** is separate from the name of the field. When you name a field, you are limited to letters, numbers and spaces only, and when you add a field to a form, the SmartList To Go handheld app automatically creates a label based on the field's name. Once created, however, you are free to change the static text to create user-friendly labels that use any combination of letters, number, spaces, punctuation and special characters.

Static Text

You can also create static text on lines in the form that do not contain a field, perhaps as a heading to a group of fields. Static text is essentially a label that extends across both columns of the form, the left column where labels usually reside and the right columns where the values normally go. In all other ways, static text behaves just like a label, except that if the text is longer than the screen is wide, static text will continue on additional lines. Field labels are restricted to one line only.

3.3.3.3 Closing Records

To close a record and return to the View List screen, click the **OK** button to return to the View List and save any changes you have made to the record. Click the **Cancel** button to return to the View List and abandon any changes you have made to the record.

3.3.4 Closing a SmartList

To close a SmartList from the List View screen, click on the **Home icon** located in the bottom left hand corner of your screen. This will display the Home screen and its current list of SmartLists. There is no need to manually save your changes when leaving a SmartList; the SmartList To Go handheld application will save your changes automatically.

This completes our tour of the SmartList MyBooks. You should now have a firm understanding of SmartList To Go's basic theory of operation and a familiarity with much of the vocabulary you will need to use the SmartList To Go handheld app. As you continue to the next chapter, you will begin to learn how to make your own SmartLists to fit your needs.

4. Designing SmartLists

Now that Chapter 3 has familiarized you with SmartList To Go and its vocabulary, this chapter will teach you how to design and create your own SmartLists.

In This Chapter . . .

- **4.1 SmartList Creation**
 - **4.1.1 Creating a SmartList**
 - **4.1.2 Setting SmartList Properties**
 - **4.1.3 Using the SmartList Main Menu**
 - **4.1.4 Renaming a SmartList**
 - **4.1.5 Deleting a SmartList**
- **4.2 Designing a SmartList**
 - **4.2.1 The Field Editor Screen**
 - **4.2.2 The Form Designer Screen**
 - **4.2.3 The View Editor Screen**
 - **4.2.4 The List Editor Screen**
- **4.3 Records**
 - **4.3.1 Creating Records**
 - **4.3.2 Duplicating Records**
 - **4.3.3 Deleting Records**
 - **4.3.4 Deleting All Records**

4.1 SmartList Creation

This section will discuss how to create a SmartList and configure some of its properties, as well as how to use the Main Menu in many of the SmartList To Go handheld app's most common screens.

In This Section . . .

- 4.1.1 Creating a SmartList
- 4.1.2 Setting SmartList Properties
- 4.1.3 Using the SmartList Main Menu
- 4.1.4 Renaming a SmartList
- 4.1.5 Deleting a SmartList

4.1.1 Creating a SmartList

To create a SmartList, begin from the SmartList To Go handheld application Home screen or a SmartList View List screen and follow these steps.



SmartList Tools About			
Create New SmartList... /Y			
artist	▼	3	3/23
BigShop	▼	268	3/14

1. Tap on the New SmartList icon to open the Create New SmartList screen. You can also click on the SmartList To Go tab at the top of the screen to open the **Main Menu** and select **SmartList > Create New SmartList** or use the shortcut /Y.
2. Enter the **Name** of the SmartList in the field provided.
3. Enter the **Location** of where the SmartList will reside (either Handheld or Card, if available).
4. Apply a **Category**, if desired.
5. Choose from one of the three **Creation Methods** provided and select the **Next** button:
 - Use SmartList Wizard. The SmartList Wizard allows you to enter fields in a simplistic manner. In the **Name** field enter a name for the field, such as "Title", and then choose the **Type** of field that it is, such as Text. Select the **Add** button and the field is added. Repeat the process for all fields that you wish to add. When finished adding fields, select the **Next** button.
 - Create from Template. This creation method allows you to use an existing SmartList as a starting point for a new SmartList. From the Choose List dropdown menu, simply choose an existing SmartList to be used as your starting point. After selecting a SmartList, you will also have an option to import the data from that SmartList.
 - Enter Data Directly. This option opens the new SmartList directly into the Field Editor. You can add fields by selecting the **New** button in the Field Editor.

4.1.2 Setting SmartList Properties

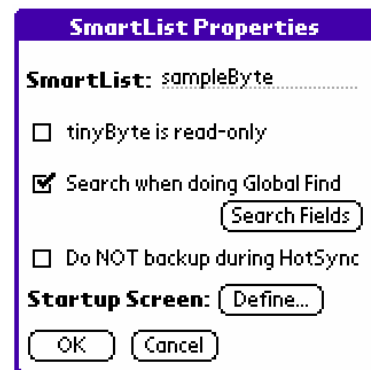
A SmartList's properties control preferences that will apply to the entire SmartList. Control these preferences using the SmartList Properties screen.

Set SmartList properties when you create the SmartList, or, [to open the Properties screen](#), click on the tab at the top of the page to open the **Main Menu** and select **SmartList > SmartList Properties**. The Properties screen will open and display the following controls:

SmartList is Read-only

Check this box to prevent any changes, additions or deletions to the SmartList. This will protect the current data in the SmartList by disabling your ability to change, add or delete information. However, you will still be able to make design changes effecting field and form design; this allows you to change how the data appears without changing the data itself.

By default, the SmartList is read-only checkbox is not checked; you must check it manually. Note that this is not usually a useful option to select when you first create the SmartList. It is far more useful to select after you have entered your data.



[To convert a SmartList to read-only](#), begin from the View List screen or a Record screen and follow these steps:

1. Click on the tab at the top of the screen to open the **Main Menu** and select **SmartList > SmartList Properties**. This will open the SmartList Properties screen.
2. Click to check the **SmartList is read-only box**.
3. Make any other changes to the SmartList properties you require and click **OK** to return to the previous screen.

The SmartList is now read-only. You may alter fields and forms, but you may not change, add or delete data.

In Section 5.9 Security you will learn how you can protect your data further with security settings that will prevent users from unchecking the read-only option without the proper password. This is particularly useful for SmartLists containing sensitive data you intend to share among a number of users and handhelds.

Search when doing Global Find

Check this box to include fields from this SmartList in SmartList To Go's Global Find function. The Global Find enables you to search for words, phrases and other field values across all existing SmartLists.

To include a SmartList in all Global Finds, begin from the View List screen or a Record screen and follow these steps:

1. Click on the tab at the top of the screen to open the **Main Menu** and select **SmartList > SmartList Properties**. This will open the SmartList Properties screen.
2. Click to check the **Search when doing Global Find box**. The Search Fields button will appear when you check the box.
3. Click the **Search Fields** button to open the Field Find Properties screen. You will use this screen to indicate which fields to search during the Global Find and which fields to ignore.
4. Check the boxes next to the names of all fields you wish to search during a Global Find. You may quickly include all fields by clicking the **All** button near the lower right corner of the screen. You may quickly remove all fields from the Global Find by click the **None** button in the lower right corner. Removing all fields effectively removes the entire SmartList from Global Finds.



Note: You cannot configure a SmartList to be included in Global Finds until you have at least designed the fields that you wish to include. If the fields do not yet exist, you will not be able to select them.

5. Click the **OK** button to save your selections and return to the SmartList Properties screen, or click the **Cancel** button to abandon your changes.
6. Make any other changes to the SmartList properties you require and click **OK** to return to the List View or Records screen.

All Global Finds will now search this SmartList, looking only in those fields you selected. If no fields are selected, or the Search when doing Global Find box is not checked, Global Finds will ignore this SmartList.

Startup Screen: Define

Use the Startup Screen: Define button to create a memo screen called "About this SmartList" that will open automatically when you open the SmartList. You can use the About this SmartList screen to serve as a title page, provide a brief description of data contained within, to remind other users how to perform key tasks with the SmartList, or even to leave yourself notes for the next time you open that SmartList.

To create a startup screen for your SmartList, start from the SmartList Properties screen and follow these steps. You can begin this process when the SmartList Properties screen opens automatically during SmartList creation, or you can access this screen from the List View screen or a Record screen by clicking on the tab at the top of the screen to view the **Main Menu** and selecting **SmartList > SmartList Properties**.

1. Click on the **Startup Screen: Define** button to open the About this SmartList screen.
2. Enter the desired text on the lines provided.
3. Check the box next to Show this screen at startup to activate this startup screen.
4. Click the **OK** button to return to the SmartList Properties screen.

The Startup Screen is now defined and will appear each time you open the SmartList.

To prevent a Startup Screen from continuing to display, simply uncheck the box next to Show this screen at startup. You may do this when the screen opens automatically, or access the screen at any time as described above. It is not necessary to change any other aspect of the screen's configuration to disable it.

4.1.3 Using the SmartList Main Menu

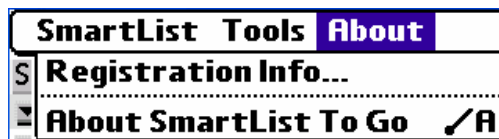
The SmartList Main Menu provides you with a primary interface for performing major functions and accessing important screens in the SmartList To Go handheld application. It is located at the top of the View List screen or a Record screen. To access the **Main Menu**, click on the tab at the top of the screen.

The exact options present in the Main Menu depend upon what screen you view it from; some actions can't be done or don't make sense from all starting points. Below you will find a general synopsis of the options you will find in some of the major screens.

In the SmartList To Go Home Screen

The Main Menu in the SmartList To Go handheld application Home screen contains three pull-down menus: SmartList, Tools, and About.

- **SmartList:** The SmartList pull-down menu contains the command for creating a new SmartList, for searching for existing SmartLists, and for Zooming to see more or less text on screen (on supported devices).
- **Tools:** The Tools pull-down menu contains commands for setting SmartList To Go Record View preferences, color preferences, Plug-in Manager, and Beaming.
- **About:** The About pull-down menu contains commands for registering your copy of SmartList To Go and accessing version information.

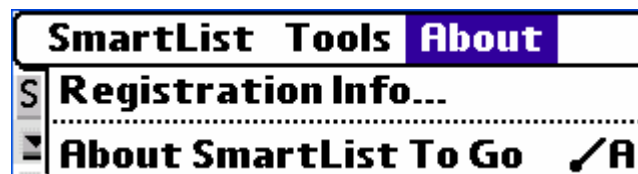
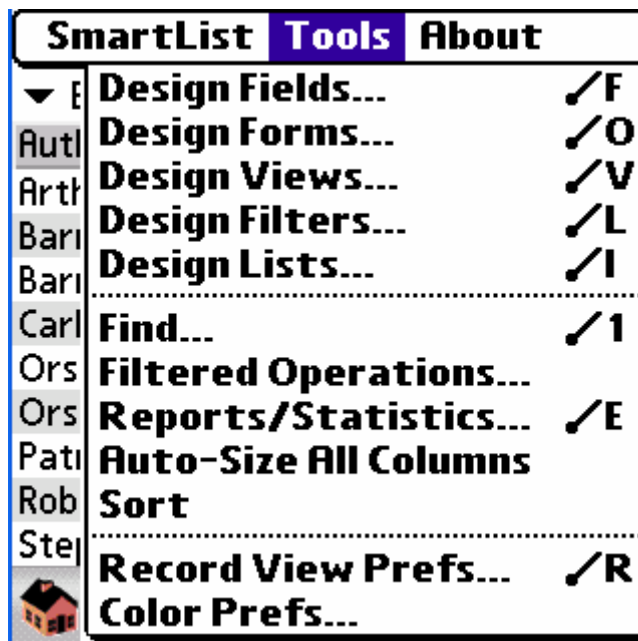


In the SmartList List View Screen

The SmartList To Go Home Main Menu

The Main Menu in the SmartList To Go handheld application Home screen contains three pull-down menus: SmartList, Tools, and About.

- **SmartList:** The SmartList pull-down menu contains commands for creating a new SmartList, deleting the SmartList or individual records, configuring SmartList properties, adding security measures, beaming and printing or exporting reports.
- **Tools:** The Tools pull-down menu contains commands for accessing the design screens, including screens for design:
 - Fields
 - Forms
 - Views
 - Filters
 - Lists



The List View Main Menu

This menu also includes commands for performing filtered operations, generating reports, auto-sizing columns, sorting, and configuring SmartLists preferences.

- **About:** The About pull-down menu contains commands for registering your copy of SmartList To Go and accessing version information.

In a Record Screen

The Main Menu in the Record screen contains three pull-down menus: Actions, Edit, and Options.

- **Actions:** The Actions pull-down menu contains the commands for creating and duplicating, deleting, printing, exporting or beaming records.
- **Edit:** The Edit pull-down menu contains commands you would expect in the Edit menu of a Windows application, including cut, copy and paste commands, Undo and Select All. It also contains a command for calling up help with the graffiti interface.
- **Options:** The Options pull-down menu contains a command for switching the alignment of field labels from left to right aligned or vice versa.

Actions	Edit	Options
New Record	✓N
Duplicate Record	✓J
Delete Record...	✓D
Print / Export Layout...	
Print Record	✓W
Export Record	✓M
Beam Record	✓B

Actions	Edit	Options
Title:	Undo	✓U
Author:	Cut	✓X
	Copy	✓C
General	Paste	✓P
ISBN:	Select All	✓S
Publ:	Keyboard	✓K
Pages:	Graffiti Help	✓G
Edition:		

Actions	Edit	Options
Tit	Switch Alignment	✓H

The Record Main Menu

In Most Other Screens

The Main Menu in most other screens contains at least an Edit menu. The Edit pull-down menu contains commands you would expect in the Edit menu of a Windows application, including cut, copy and paste commands, undo and select all. It also contains a command for calling up help with the graffiti interface.

Edit
Undo
Cut
Copy
Paste
Select All
Keyboard
Graffiti Help

The Main Menu in most other screens

4.1.4 Renaming a SmartList

Rename a SmartList using the Properties screen. You can access this screen using the corresponding SmartList pull-down menu from the SmartList To Go handheld application Home screen or within the SmartList itself.

To rename a SmartList from the SmartList To Go Home screen, follow these steps:

1. Open the pull-down menu corresponding to the SmartList you wish to rename and select **Rename**. This will open the SmartList Properties screen.
2. Enter the new name in the space provided.
3. Click **OK** to return to the SmartList To Go handheld application Home screen.

-OR-

To rename a SmartList from the SmartList's List View screen, follow these steps:

1. Click on the tab at the top of the screen to view the **Main Menu** and select **SmartList > SmartList Properties**. This will open the SmartList Properties screen.
2. Enter the new name in the space provided.
3. Click **OK** to return to the SmartList List View screen.
4. The SmartList is now renamed.

4.1.5 Deleting a SmartList

To delete a SmartList, go to the SmartList To Go handheld Home screen and follow these steps:

1. Open the pull-down next to the SmartList you wish to delete.
2. Select Delete. This will remove the SmartList from the SmartList To Go handheld app.

-OR-

Start from the List View screen in the SmartList you wish to delete. Click on the tab at the top of the screen to open the **Main Menu** and select **SmartList > Delete This SmartList**. This will permanently remove the SmartList from SmartList To Go and return you to the SmartList To Go handheld application Home screen.



Warning: Be very careful when deleting a SmartList. SmartList deletions cannot be undone if you change your mind or accidentally select the wrong SmartList.



Notes: When you delete a SmartList, it is completely removed from your handheld. The SmartList's name may still appear in the Recently Used SmartLists pull-down menu located on the Name tab at the top of the screen. However, if you attempt to access the SmartList using this menu, an error message will indicate that the SmartList no longer exists.

4.2 Designing a SmartList

The exact process of designing a SmartList will always depend upon the task for which you create the SmartList. However, creating every SmartList will, at the minimum, include these three steps:

1. Planning
2. Designing fields
3. Designing the form

In many cases, you will perform additional steps to further customize and enhance your SmartList's effectiveness, but when these three steps are complete, you will be ready to begin entering data and using your SmartList.

As you design your SmartList, you will use a number of different design and configuration screens, but your three primary interfaces will be:

- The Field Editor screen
- The Form Designer screen
- The View Editor screen

This section will describe each of these design screens in detail.

In This Section . . .

- **4.2.1 The Field Editor Screen**
- **4.2.2 The Form Designer Screen**
- **4.2.3 The View Editor Screen**

A Note on Planning

Form follows function. This old saying holds true especially in an application as straight forward and direct as the SmartList To Go handheld application. Before you begin any of the hands-on aspects of designing your new SmartList, think about the function you would like it to perform and how that function effects the answers to the following questions:

- What information will I need?
- What kinds of information will I need?
- What is the best way for me to enter this information?
- What is the best way to display this information?
- Under what circumstances will I review this information, and how does that effect the display design?
- How often will I change information I have already entered and how will changes effect the whole?

This is by no means a complete list of the questions you will have to ask yourself before you begin, but the answers will help you begin to shape in your mind how your SmartList should look.

The questions above are very general, but when you consider them, try to make them as specific as possible. For example, if you are creating a SmartList to manage contact information, how will you decide to enter the contact's name?

There are certainly many choices. You could:

- enter the full name in a single text field -OR-
- enter the last name first followed by the first name in a single text field -OR-
- enter the first and last names in separate fields.

Which should you choose?

The answer might be effected by whether these are business or personal contacts. You are more likely to look up a business contact by their last name, suggesting the last name first method. On the other hand, you are more likely to look up a personal contact by their first name, suggesting that the first name first would be more effective.

Perhaps either is likely, in which case you would enter the first and last name in separate fields so you could sort lists according to either.

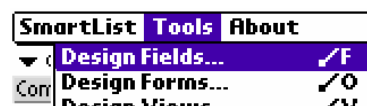
Perhaps none of these options is best, because you are more likely to look a contact up referencing the company they represent.

Of course, the more SmartLists you create, the easier it will be to make useful SmartLists quickly. In the beginning, you may find it useful to plan out the fields and entry forms on paper before continuing.

4.2.1 The Field Editor Screen

The Field Editor screen is your primary interface for creating, configuring, altering or deleting fields.

To open the **Field Editor** screen, open the **Main Menu** and select **Tools > Design Fields**.



Opening the Field Editor screen

In This Section . . .

- 4.2.1.1 Creating Fields
- 4.2.1.2 Field Types
- 4.2.1.3 Deleting Fields

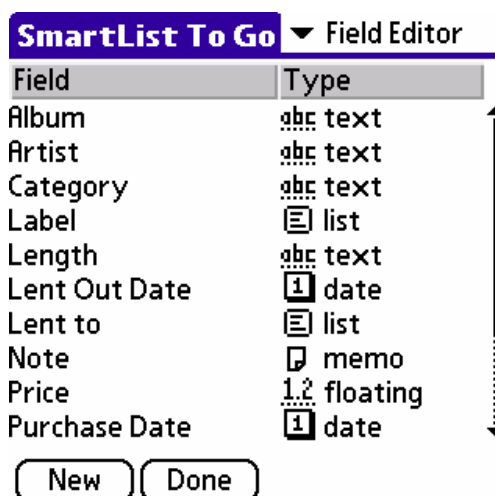
4.2.1.1 Creating Fields

To create your fields, start from the Field Editor screen and follow these steps:


1. Select the **New** button to open the **Single Field Editor**.

2. Enter a **Name** for the new field and select a **Type** for the field. For each field type you will

notice a description of the field type in the box below **Type**. For a description of each field type, see Section 4.2.1.2 Field Types.



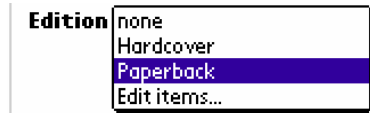
The Field Editor screen

3. In the **Single Field Editor**, click on the Field Properties button  on the right side of the field type selected to open the Properties screen for that field type (if applicable). The controls in the Field Properties screen will vary depending upon the field type's characteristics. Some field types will not have a Field Properties button because they have no configurable characteristics. Configure the field's properties and return to the Single Field Editor screen.
4. Repeats Steps 1, 2 and 3 to continue creating as many fields as you require.
5. When you have created your fields, click the **Done** button in the bottom of the screen.

You have now created your fields. Next you need to design the form.

4.2.1.2 Field Types

SmartList To Go currently accommodates 22 field types. Below you will find brief descriptions of each type. For more complete descriptions, including configurable properties, see Appendix A.

- **Text** The Text field type stores any combination of letters, numbers, punctuation marks and spaces up to 254 characters. Text stores and recalls the data literally.
- **Integer** The Integer field type stores numerical values that are positive or negative whole numbers ranging from -32768 to +32767. The Integer field type does not allow decimals.
- **Long** The Long field type is identical to the Integer field type except that it accepts a much larger range of values, from -2,147,483,648 to +2,147,483,647. In order to provide this increased range, the Long field type devotes twice as much memory to storing values as the Integer type. Therefore, unless the increased range is specifically required, it is usually far more efficient to select Integer.
- **Float** The Float field type stores numeric values with up to three digits after the decimal point.
- **List** The List field type stores a value from a limited list of choices you create. The list appears as a pull down menu of the choices you created. See the example, taken from the sample SmartList MyBooks, to the right.
- **Checkbox** The Checkbox field type stores a true or false value depending on whether or not the box is checked.
- **Date** The Date field stores dates including day, month and year.
- **Time** The Time field stores times including hour, minute and second.
- **Expression** The Expression field stores a value determined by a mathematical expression that uses one or more values from numerical, date or time fields as operands.
- **Advanced Expression** The Advanced Expression field stores a value determined by a mathematical or boolean expression that uses constants and one or more values from other fields as operands. In addition to calculating numerical values, this field can also concatenate text value, including converting a mathematical value to text automatically in order to concatenate with another text value.
- **Memo** The Memo field type stores longer text entries up to 4096 characters long. Entries may include letter, numbers, punctuation, special characters and spaces.
- **Key** The Key field stores an uneditable integer value that, starting with 1, automatically increments as records are added to the SmartList. Adding a Key field creates a unique identifier for each record that guarantees that every record remains unique even if the value in all the other fields are identical.
- **DB Join** The DB Join field enables you to inter-relate SmartLists. Rather than storing a value directly, this field contains a pointer to another field contained in another SmartList. The DB Join field will return the value of the field it points to almost as though the value was present in the same SmartList.
- **DB Lookup** The DB Lookup field works together with the DB Join field to increase the inter-relation between SmartLists. Once you configure the DB Join field to

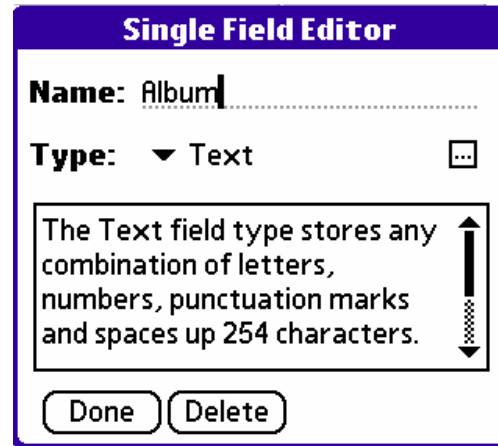
Sample List field type

point to a certain field in another SmartList, you can configure one or more DB Lookup fields to point to other fields in the same record and return corresponding values. You can even use DB Lookup fields in expressions and conditional expressions.

- **One to Many** The One to Many field links a single SmartList record to two or more records in another SmartList. As an example, consider that you created a SmartList named Authors to coincide with the sample SmartList we have already referred to, MyBooks. A One to Many field in Authors could provide a link to all the records in MyBooks that described books by that author. You could click on the One to Many field in the record for Stephen King and see a list of the records including "Bag of Bones," "Carrie" and "Hearts in Atlantis."
- **Address Join** The Address Join field enables you to relate a SmartList to the Address Book application built in to the Palm OS. An Address Join works in the same way a DB Join relates one SmartList to another. It creates a pointer to a field contained in the Address Book, adding that value to the SmartList record.
- **Address Lookup** The Address Lookup field works with the Address Join field, in the same way DB Lookup works with DB Join, to increase the inter-relation between the SmartList and the Address Book application. Once you have configured the Address Join field, you can configure one or more Address Lookup fields to extract other items of information from Address Book records.
- **Radio Button** The Radio Button field allows you to create a list of up to eight previously defined selections, only one of which may be selected at a time.
- **Image** The Image field inserts an image into the record in one of four formats. These formats are: black and white; 4 shade gray-scale; 16 shade gray-scale; 256 colors.
- **Conditional Expression** The Conditional Expression field evaluates a boolean expression you create. You configure the field to return one value if the expression evaluates to true and another value if the expression evaluates to False.
- **Auto Incremental** The Auto Incremental field is an integer field. The first record you create will automatically store a value of 1 in this field. Each subsequent record will continue to count upwards, adding 1 to the value saved in the previously created field. This is very similar to the Key field type, but be aware that you can manually change the value in an Auto Incremental field while the value of a Key field is not changeable. Therefore, only the Key field guarantees you a unique identifying field for each and every record.
- **SmartList To Go Plug-In** The SmartList To Go Plug-In field enables you to expand the functionality of SmartList To Go with special plug-ins. The exact functionality of this field depends upon the plug-in inserted. For information on using specific plug-ins, see www.dataviz.com/smartlisttogo.

4.2.1.3 Deleting Fields

To delete a field, tap on the field name or type to open the **Single Field Editor**. Select the **Delete** button to delete the field.



The Single Field Editor screen

4.2.2 The Form Designer Screen

The Form Designer screen is your primary interface for configuring your SmartList's input and display form.

To open the Form Designer screen from the Field Editor screen, open the pull-down menu in the upper right corner of the screen and select **Form Designer**.

To open the Form Designer screen at any other time, open the **Main Menu** and select **Tools > Design Forms**.

In This Section . . .

- **4.2.2.1 About Form Layout**
- **4.2.2.2 Adding Fields**
- **4.2.2.3 Moving and Changing Fields**
- **4.2.2.4 Renaming a Field Label using the Label Properties Screen**
- **4.2.2.5 Removing Fields**
- **4.2.3.0 Changing Column Layout**
- **4.2.2.7 Changing Label Alignment**
- **4.2.2.8 Using Tabs**
- **4.2.2.9 Creating a Default Form**
- **4.2.2.10 Clearing a Default Form**

4.2.2.1 About Form Layout

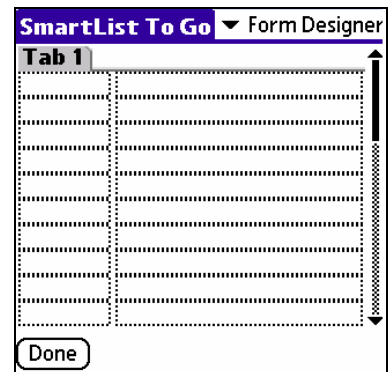
When first opened, the Form Designer screen will display an empty form. To create your SmartList's input and display form, you will add the fields you created. Each line on a form can contain no more than one field and its label. The form labels appear in the left column of the form; you enter data into each field in the right column of the form.

There are a number of different ways you can customize your form:

- Adding, moving and removing fields
- Changing fields labels or adding labels to blank lines
- Adding, removing and naming multiple tabs. Adding multiple tabs is like creating multiple pages for your form.
- Adjusting form layout by moving the divider between the columns left or right.
- Adjusting form layout by moving the tab separator up or down. Fields placed above the tab separator will always appear on the form, no matter what tab you select; fields placed below



Opening the Form Designer screen



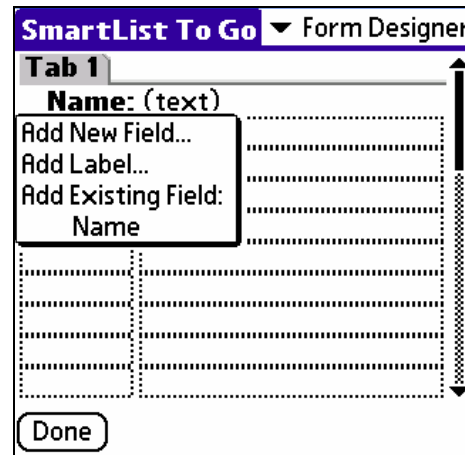
A Blank Form with the Default Layout

the separator will change depending on the tab selected.

4.2.2.2 Adding Fields

To add a field to the form, follow these steps:

1. Click on the form line on which you would like the field to appear. This will open a pull-down menu with options to Add New Field, Add Label, Add Existing Field (with existing fields listed).
2. Select the name of the field you wish to add, or select **Add New Field** to open the **Single Field Editor** to create a new field.



The Field Editor screen

The field is now added to the form. The field label will appear on the left column of the form, and the field's type will appear in parenthesis on the right column of the form. When you leave the Form Designer screen and begin to use this form for data entry, a blank line will appear in the right column instead of the field type. This blank line is where you will enter data.

The field label will default to the field's name, but you can change the label to suit your needs without altering the field name itself. Section 4.2.2.4 Renaming a Field Label below describes how to change field labels.

4.2.2.3 Moving and Changing Fields

To move a field from one line to another on the form, click on the field label and drag with the stylus to the desired line. You may move a field in this manner to any line that does not already contain a field.


To change the field in a specific line, first remove the existing field, then add the new field in its place, using the directions for removing a field in Section 4.2.2.5 Removing Fields and the directions for adding a field in Section 4.2.2.2 Adding Fields.

4.2.2.4 Renaming a Field Label using the Label Properties Screen

When you add a field to a form, the SmartList To Go handheld app will create a label that is the same as the field name by default. Once this is done, however, you can rename the field label and customize its display in a number of ways using the Label Properties screen.

To rename and customize a field label, start from the Form Designer screen and follow these steps:

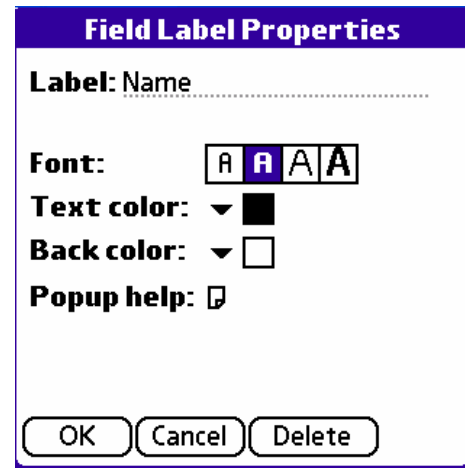
1. Click on the name of the field you wish to modify and choose Label Properties from the popup menu. This will open the **Field Label Properties** screen.
2. Enter the desired label in space marked Label. The field name, which appears on the line above, will not change in this process. (You must rename fields in the Field Editor screen.)
3. If desired, select a custom font from the four choices under Font.
4. Select the label alignment style using the Alignment control. Labels may be left aligned, centered, right aligned or justified.
5. If desired, select custom text and background colors using the Text color and Back color menus. Click on the down arrow to open a Palette screen, select the desired color and click **OK**.
6. If desired, add Popup help to a field by clicking on the page icon next to Popup Help. This will open a Popup Help screen. Enter the help text exactly as you would like it to appear, as in the example to the right.

This will add a Pop-up Help icon  to the field when you are finished. When you click this icon, the Popup help screen will open and display the text exactly as you entered it.

Click **OK** to return to the Label Properties screen.

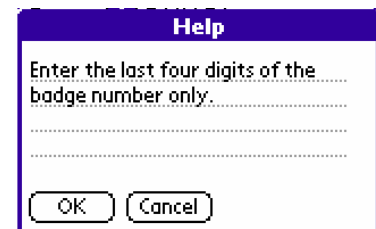
6. Click **OK** to complete the Field Label change.

The field label is now changed and modified.



The screenshot shows the 'Field Label Properties' dialog box. It has a title bar 'Field Label Properties'. Inside, there is a 'Label: Name' field with a dotted line for text entry. Below this are four font style icons: a standard 'A', a bold 'A', an italic 'A', and a bold italic 'A'. The 'Font:' label is to the left of these icons. Below the font icons are two color selection controls: 'Text color:' with a black square and 'Back color:' with a white square. Below these is a 'Popup help:' checkbox which is checked. At the bottom are three buttons: 'OK', 'Cancel', and 'Delete'.

The Label Properties screen



The screenshot shows the 'Help' dialog box. It has a title bar 'Help'. Inside, there is a text entry field with the prompt 'Enter the last four digits of the badge number only.' followed by a dotted line. Below the text field are two buttons: 'OK' and 'Cancel'.

The Popup Help screen

4.2.2.5 Removing Fields

To remove a field from a form, tap on the field name and choose Remove Field from the popup menu.

4.2.3.0 Changing Column Layout

You can adjust the column widths to accommodate larger labels or provide more room for data entry.

To change column layout from the Form Designer screen, follow these steps:

1. Click on the tab at the top of the screen to open the **Main Menu**, and select **Options > Resize Labels**. A vertical line will appear to indicate the separator between the two columns.
2. Click on this line and drag to the left or right to change the columns to the desired widths. The vertical line will disappear automatically when you lift the stylus.

-OR-

If you feel you do not need the vertical line to indicate the separator between the columns for you, simply click where the vertical line would appear and drag to the left or right to change the columns to the desired widths. This method will resize the columns even though you did not use the menu command as described above, however, using the menu command is often easier because it provides that vertical line as a visual indicator.

4.2.2.7 Changing Label Alignment

You can display labels either right aligned or left aligned.

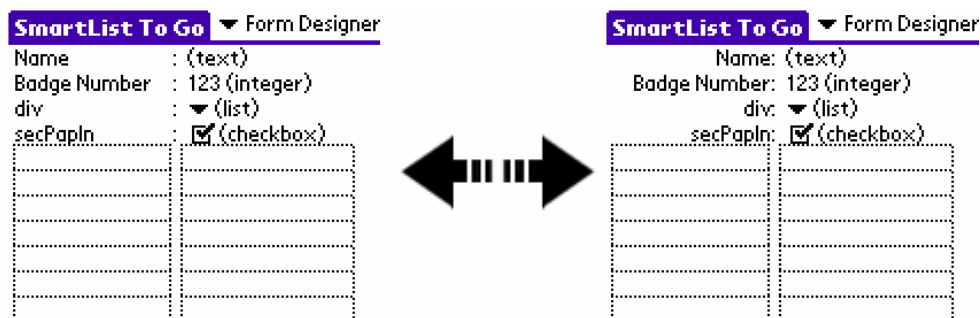
To change label alignment from left aligned to right aligned or vice versa, start from the Form Designer screen. Click on the tab at the top of the screen to open the **Main Menu**, and select **Options > Switch Alignment**. This will change the alignment of labels from left aligned to right aligned and vice versa.



The Resize Labels menu command



The Switch Alignment menu command



4.2.2.8 Using Tabs

Tabs allow you to create forms with multiple pages. Take a look at the records in the sample SmartList, MyBooks, for an example of how tabs work. Each tab in the form (pictured at right) contains different fields. Select which tab to show by clicking on the tab name.

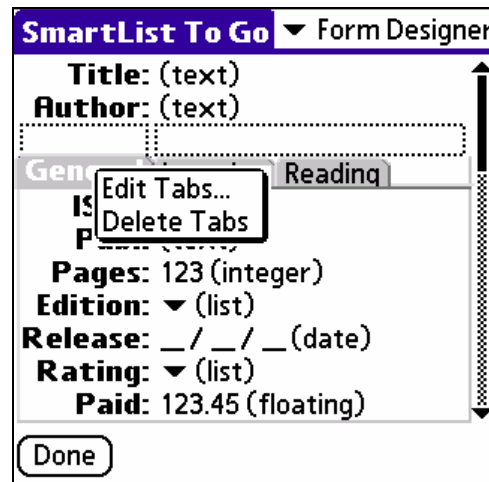
You can create tabs in your forms as well. Tabs increase the number of fields that fit on a form and enhance usability by grouping fields together logically.

First you will decide how you wish to group your fields. Next you will create and name the tabs. Lastly you will add the fields to the appropriate tabs as described in Section 4.2.2.2 Adding Fields above. The fields on each tab only display when that tab is selected. In this manner, you can create up to 14 tabs.

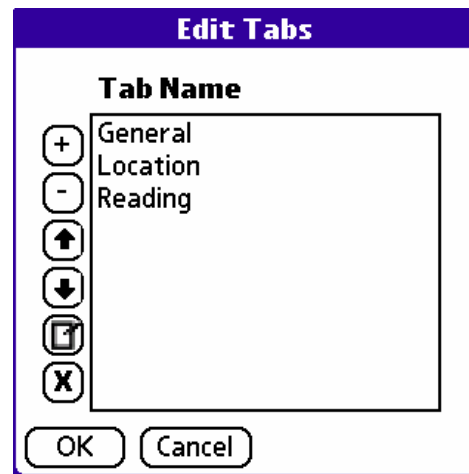
Fields you want to display all the time, regardless of the tab selected, you put above the tabs. You can adjust the tab layout upward or downward to accommodate more or fewer fields above the tabs.

To create or modify tabs in the Form Designer screen, you must first plan out how you will group your fields on to tabs, how many tabs you will need and what you will name the tabs. Then tap on an existing **Tab** and choose **Edit Tabs...** from the pull-down menu. You can modify the tabs from this screen as follows:

1. To add a tab, click on the **plus** button and enter a name for the tab in the **Add Tab** screen.
2. To delete a tab, highlight the tab name you want to delete and choose the **minus** button.
3. To reorder tabs, highlight a tab and choose the up or down arrow to reorder the tab accordingly.
4. To rename a tab, highlight the tab you want to rename, and choose the rename symbol below the down arrow. Enter the new name.
5. To delete all tabs, choose the **X** button. You may also delete all tabs in the Form Designer by tapping on a tab and choosing Delete Tabs from the popup menu.



The Edit Tabs Pop-up menu



The Edit Tabs screen

4.2.2.9 Creating a Default Form

The quickest method of creating a form is to use the Generate Form Function. This function is available in the Main Menu in the Form Designer screen and it creates a basic, utilitarian form with the following properties:

- All existing fields will appear on the form in the order in which they are listed in the Field Designer. If multiple tabs are required to accomplish this, the correct number of tabs will be created.
- The names of the fields indicated in the Field Designer will be used as field labels. Field labels will appear in bold-face type and field values will display in the normal font.
- Image fields will display images as icons.

To create a default form, open the Form Designer screen and follow these steps:

1. Open the Main Menu and select **Generate Form**.
2. An alert window will open stating the new form will replace the existing form. Click **OK** to continue the function or click **Cancel** to abort. If you click **OK**, SmartList To Go will automatically generate the default form.

The process is now complete.

4.2.2.10 Clearing a Default Form

To clear a form design, open the Form Designer screen and follow these steps:

1. Open the Main Menu and select **Clear**. This will open the Clear Form screen. You will use this screen to indicate what parts of the form to clear and how much of the form to apply it to.
2. Under **Remove**, select to remove the entire field or just the formatting. Removing the formatting removes any adjustments you may have made regarding fonts or color and returns the field to its default display state.
3. Under **Apply to**, select to apply this change to the entire form, to the current tab only, or to common fields. Removing the entire form will leave your tabs intact, but remove all fields on all tabs and above the tabs. Removing common fields leaves the fields on the tabs and removes only those fields which appear above the tabs.
4. Click **OK** to continue the function or click **Cancel** to abort. If you click **OK**, the SmartList To Go handheld application will clear the form according to your instructions. This function is not undoable.

The process is now complete.

4.2.3 The View Editor Screen

A View is the tabular display of records you see when you first open a SmartList. Each column represents one field in the records according to the label at the top of the column; each row represents a different record. In the same way the a form provides a quick and easy view of all the information in an individual record, the view quickly and easily lays out the most important kinds of information in every record

With the SmartList To Go handheld application, you can design up to 12 different views customized according to what information you need most in different situations, how you wish to order records on the table and even which records you want to display.

Definition: A View is a tabular display of the data contained in one or more records in the SmartList. In this table, each column represents a different field, and each row represents a different record. Views are customizable to show any fields in any combination and in any order.

In This Section . . .

- **4.2.3.1 Designing a View**
- **4.2.3.2 Deleting a View**

4.2.3.1 Designing a View

The View Designer screen is your primary interface for creating and modifying views.

To create or modify a view, follow these steps:

1. Click on the tab at the top of the page to view the **Main Menu** and select **Tools > Design Views** to open the View Designer screen.

Take a moment to note some of the key components of this screen. The View pull-down menu at the top of the screen indicates what view you are currently editing. The Column area in the center of the screen indicates what fields appear in the view and in what order; the first item in this list is the farthest to the left, the second item is next, and so on. You will use the buttons immediately to the left of the Column area to adjust this order.

2. Open the View pull-down menu and select the view you wish to create or modify.

To modify an existing view, select that view's name (By Title, in the example at right). To create a new view, select one of the unused, default names on the list.

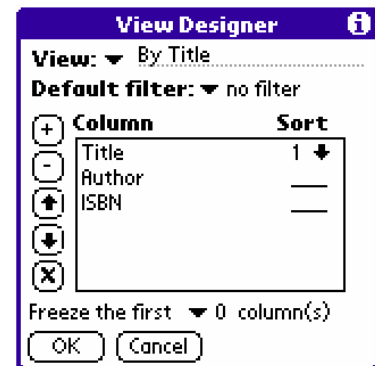
Replace the default text in View with the view's new name. Note that SmartList To Go considers that any view that does not have any fields arranged in the Column area is not in use, even if the name is changed from the default.

3. Add and arrange fields in the Column area to determine the order they appear in the finished view.

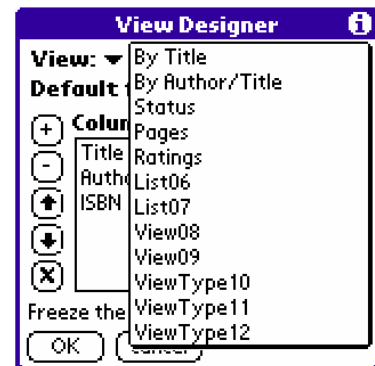
To do this, click on the **Plus** button to open a pull-down menu in which to select the desired field. Repeat this action to add each field to the view. Highlight a field and click the **Minus** button to remove it from the view. Highlight a field and click on the up and down buttons to change its position in the field order. Click the **X** button to clear the current list of fields in the Column area.

4. If you would like to sort record entries according to the data in a specific field, use the pull-down menus on the right side of the Column area to indicate your sorting preferences.

First, select the field you want to sort by and click on the corresponding pull-down menu. Select either **primary ascending** or **primary descending**. Ascending means that the view will list these records so that this field's data is in alphabetical order. Descending will sort records in reverse alphabetical order. For example, in the sample SmartList MyBooks, the view By Title lists records in alphabetical order according to the first word in the title. By Author/Title lists records in alphabetical order according to the author's name.



The View Designer screen



The View Designer screen and View menu

Next, if you wish, select a second field to use when creating secondary sorts, open the corresponding pull-down menu and select **secondary ascending** or **secondary descending**. For example, My Books may have a number of books by the author Stephen King. Selecting Author as the primary ascending sort field will group all books by Stephen King together; the books' order will be essentially random because they each have identical data in the author field. However, selecting Title as the secondary ascending sort field will then order the books by Stephen King alphabetically.

Lastly, if you wish you can sort books to a third, or tertiary level.

5. If you wish, select to freeze view columns.

This means that when you scroll left and right in the view to see the other columns, up to three of the first columns will not scroll. They will remain visible in all cases. For example, in MyBooks you could select the Title to freeze so that no matter how far you scrolled to the right, you would always know which book the data referred to.

To freeze view columns, open the **freeze the first** pull-down menu and select the number of columns to freeze, up to three.

6. Click on **OK** to apply your changes or click on **Cancel** to abandon your changes.

The View is now created and configured. Use the View pull-down menu on the View List screen to select which view to use.

4.2.3.2 Deleting a View

To delete a view configuration, follow these steps:

1. Click on the tab at the top of the screen to open the **Main Menu** and select **Tools > Form Design**. This will open the View Designer screen.
2. Open the View pull-down menu at the top of the screen and select the view you wish to delete.
3. Click on the **X** button at the bottom left of the screen to clear the fields selected in the Column area.
4. Click **OK** to apply this change and return to the View List screen.

The view is now deleted.

4.2.4 The List Editor Screen

The List Editor screen is your primary interface for configuring lists. Use this screen to configure Categories, List type fields and Text type fields with auto-writer enabled. Note that you do not create lists in this screen, you simply create and manage the items on a list; you create most lists by selecting an appropriate field type in the Field editor. The Category list always exists, even when you do not make use of it.

There are a number of ways to open the List Editor screen for any given list, depending upon what screen you begin from.

To open the List Editor screen from the List View screen, click on the tab at the top of the screen to view the **Main Menu** and select **Tools > Design Lists**. -OR-

To open the List Editor screen from the Field Editor screen or the Form Designer screen, click on the pull-down menu at the top right of the screen and select **List Editor**. -OR-

To open the List Editor screen from the Categories pull-down menu (in the List View screen or a Record screen), click on the Category pull-down menu and select the last selection, **Edit Categories**. This will open the List Editor screen with the Category list selected. -OR-

To open the List Editor screen from a field that uses a list, open the list and select the **Edit** or **Edit items** option. This will open the List Editor screen with the relevant list already selected.

In This Section . . .

- 4.2.4.1 Configuring a List
- 4.2.4.2 Using a Configured List

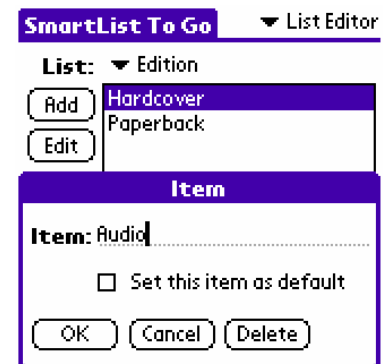
4.2.4.1 Configuring a List

Configuring a list in the List Editor screen consists of adding, removing, modifying and sorting list items.

To add a list item, follow these steps:

1. Open the List pull-down menu at the top of the screen and select the list you wish to add the item to. Lists in this menu are listed according to their field name.
2. Click on the **Add** button to add the first item to the list. This will open the Item screen.
3. Enter the text for the first item. Click **OK** on the Item screen to complete the addition or **Cancel** to close the Item screen without adding the list item.
4. Click **Done** on the List Editor screen to return to the previous screen

By default, List and Text type fields in a new record will remain empty until you specifically select a value from the list.



To modify a list item, follow these steps:

1. Open the List pull-down menu at the top of the screen and select the list that contains the item you wish to edit. Lists in this menu are listed according to their field name.
2. Select the item you wish to edit and click on the **Edit** button to modify the item. This will open the Item screen.
3. Revise the text for the item. Click **OK** on the Item screen to complete the modification or **Cancel** to close the Item screen without changing the item.
4. Click **Done** on the List Editor screen to return to the previous screen

To remove a list item, follow these steps:

1. Open the List pull-down menu at the top of the screen and select the list that contains the item you wish to edit. Lists in this menu are listed according to their field name.
2. Select the item you wish to edit and click on the **Edit** button to open the Item screen.
3. Click **Delete** on the Item screen to remove the item or **Cancel** to close the item screen without removing the item.
4. Click **Done** on the List Editor screen to return to the previous screen.

To sort a list, follow these steps:

1. Open the List pull-down menu at the top of the screen and select the list that contains the item you wish to edit. Lists in this menu are listed according to their field name.
2. Sort the list by selecting individual items and moving them up or down in the list. To do this, select the list item you wish to move and click on the **Up** and **Down** buttons as required.

To quickly sort the entire list, click on the **Sort** button. This will put the list into alphabetical order.

3. Click **Done** on the List Editor screen to return to the previous screen.

4.2.4.2 Using a Configured List

Once you have configured a list, you may quickly and easily use it from the Record screen when you enter data into a List type field or a Text type with Auto-Writer enabled or when you select a Category.

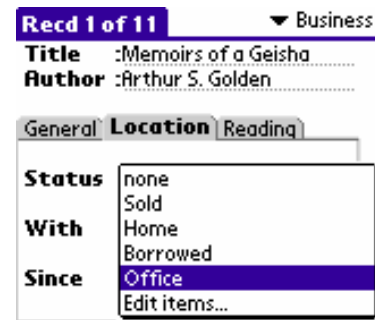
To use a configured list to enter data into a List type field, tap on the down arrow next to the field name to open the pull-down list of items. This pull-down list will contain the items you configured using the instructions of the previous section. Select the desired item to close the list and display the item in the List field.

In addition to the list items, you will also see the options **none** and **Edit Items**. Select **none** to close the list without selecting or displaying an item. Select **Edit Items** to modify the configured list, as described in the previous section.

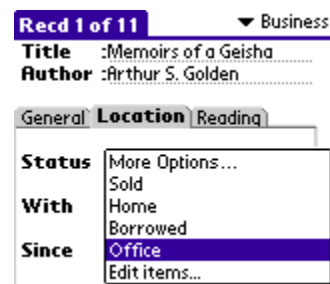
To use a configured list to enter data into a Text type field with Auto-Writer enabled, tap on the pencil icon at the far right of the field's text input line to open a pull-down list of items. This pull-down list will contain the items you configured using the instructions of the previous section. This list will also contain the options **More Options** and **Edit Items**. To select a single item, select the desired item to close the list and display the item in the Text field.

You can also select **More Options** to open the Auto-Writer screen. This screen contains your configured list with check boxes next to each item. Quickly select and display multiple list items in the same text field by checking those items you wish to display and clicking **Replace** to display only those items in the text field or clicking **Append** to add those items to the text already in the text field. When you select multiple items, the items will be separated by the character you selected in the field's properties screen. You can also select and display a single item with one tap by tapping the **Arrow** button on the right side of the screen that corresponds to your selection.

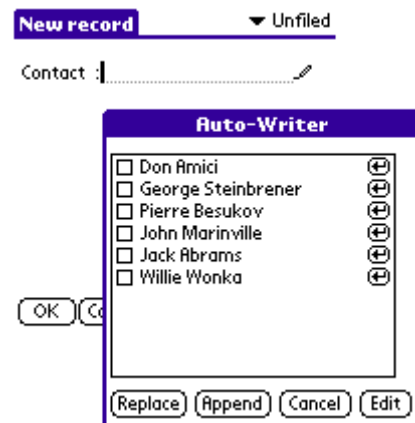
To modify the configured list, tap **Edit Items** to open the List Editor.



A Configured List for a List Type Field



A Configured List for
A Text Type Field with Auto-Writer



The Auto-Writer Screen

You'll note that Text type fields with auto-writer enabled and List type fields both work similarly. The primary difference between these two field configurations is that the List type field restricts the user to a single item chosen from the list, while the Text type field with auto-writer permits the user to select one or more list items or to manually enter an item that does not appear on the list.

To use a configured list with the Category menu, tap on the Category list located at the top right corner of the List View or Record screen. Tap the desired category to select it and close the Category list. Selecting a Category in the List View screen will change the List View so that it only contains records in that Category. Selecting a Category in the Record screen assigns that record to the selected category.

4.3 Records

Once you have created your SmartList and designed your fields and your data entry form, you are ready to begin creating records. A record is a unit of the SmartList; where the SmartList taken as a whole contains data that describes a group of items, each individual record contains the data that describes a single item.

In This Section. . .

- **4.3.1 Creating Records**
- **4.3.2 Duplicating Records**
- **4.3.3 Deleting Records**
- **4.3.4 Deleting All Records**

4.3.1 Creating Records

To create a new record, start from the List View screen and follow these steps:

1. Click on the **New Record icon** at the bottom of the screen or use the shortcut **/N**. This will open a blank record form. You designed this form when you created the SmartList.
2. Enter the appropriate data in the fields provided.
3. Click **OK** to add the new record, or click **Cancel** to abandon this record without adding it.

The new record is now created and will appear on the List View table.

4.3.2 Duplicating Records

To duplicate an existing record, start from the Record screen for that record and follow these steps:

1. Click on the tab at the top of the screen to view the **Main Menu** and select **Actions > Duplicate record** or use the shortcut **/J**. This will open a new record form and automatically populate the fields with the same data as the original record.
2. If you wish to change one or more fields, do so.
3. Click **OK** to add the new duplicate record. Click **Cancel** to close the duplicate record without saving it.



Note: Key fields are always unique. If you duplicate a record that contains a key field, the value in the field in the twin records WILL NOT be identical. The Key field in the new record will increment to the next integer just like when you create a new record by the ordinary means.

Definition: A **record** is a group of one or more fields and their corresponding values that describe one item in the group represented by the SmartList. A SmartList may include up to 65,000 records. A record may contain up to 80 fields.

4.3.3 Deleting Records

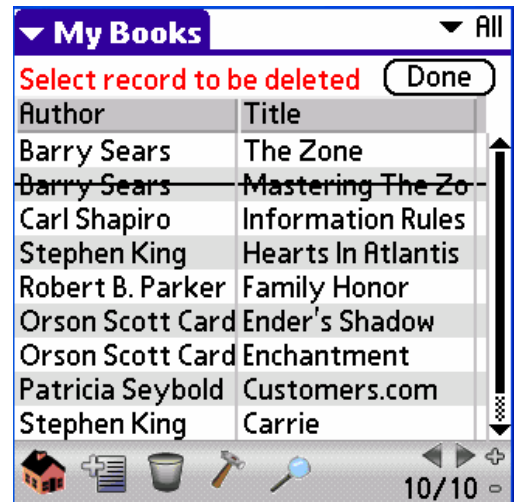
To delete one or more records, start from the List View screen and follow these steps:

1. Click on the **Delete Record icon** at the bottom of the screen. A message that asks you to click the Done button when you are done deleting will appear at the top of the screen.
2. Click on the record you wish to delete. Then click the **Done** button. This will open a menu that allows you to continue deleting the record or cancel the deletion.
3. Select **Delete** (trash can) to complete the deletion, and the SmartList To Go handheld app will delete the record and remove it from the List View table. Select **Cancel** to abort the deletion.
4. Repeat Steps 2 and 3 as often as necessary to delete all the records you wish to erase.

-OR-

Start from the Record screen in the record you wish to delete and click on the tab at the top of the screen to open the **Main Menu** and select **Actions > Delete Record** or use the shortcut **/D**. This will erase the current record, close it and return you to the View List screen.

The Records are now deleted from your SmartList.



4.3.4 Deleting All Records

You can delete all the records in a SmartList without deleting the SmartList itself. This allows you to start with a clean slate without losing the design and configuration work you have already accomplished.

To delete all records in a SmartList without deleting the SmartList, start from the View List screen and click on the tab at the top of the page to open the **Main Menu** and select **Delete All Records**. This will erase all the records from your SmartList. There is no shortcut for this process.

5. Using Advanced Functions

In the last chapter, we discussed the basic functions that will quickly get you up and running with SmartList To Go, but SmartList To Go offers a great deal of additional functionality to help you make the most of your databases. This chapter will discuss many of these more advanced functions and explain how to use them.

In This Chapter . . .

- **5.1 Categories**
- **5.2 The Expression Field**
- **5.3 The Conditional Expression Field**
- **5.4 The Advanced Expression Field**
- **5.5 Beaming**
- **5.6 Printing and Exporting**
- **5.7 Cloning a SmartList**
- **5.8 Filters**
- **5.9 Filtered Operations**
- **5.10 Security**
- **5.11 The Image Field**

5.1 Categories

You will remember from the tour of MyBooks in Chapter 3 that there is a Category pull-down menu located in the upper right corner of the View List screen. A **category** is a built-in tool for managing data more efficiently. Using this menu, you can quickly create record categories, assign records to those categories and display lists based upon those categories.

Categories are a way of grouping similar records together. You can use categories to quickly view limited lists that only include records of a certain kind. If you are already familiar with the standard PalmOS applications, like Address Book, then you are already familiar with how categories work; the SmartList To Go handheld application's categories are just like those used in PalmOS applications.

For example, you might create a SmartList to keep track of your collection of books by Stephen King with three categories: Have, Need and the built-in category All. Place any books you already own in the Have category and those you do not in the Need category. By selecting a category from the List View screen, you can quickly see a list of Stephen King books that shows only those books you have, only those you need to get, or, by selecting the built-in choice All, King's entire omnibus.

This section will provide detailed instruction on using categories.

In This Section . . .

- 5.1.1 Creating Categories
- 5.1.2 Assigning Records to a Category
- 5.1.3 Displaying by Category

The Category pull-down menu



5.1.1 Creating Categories

Use the List Editor screen to configure your list of available categories. Section 4.2.4 The List Editor Screen provides a full explanation of how to use the List Editor. This section will provide only the information that applies to Categories.

There are a number of ways to open the List Editor screen for any given list, depending upon what screen you begin from.

To open the List Editor screen from the List View screen, click on the tab at the top of the screen to view the **Main Menu** and select **Tools > Design Lists**. -OR-

To open the List Editor screen from the Field Editor screen or the Form Designer screen, click on the pull-down menu at the top right of the screen and select **List Editor**. -OR-

To open the List Editor screen from the Categories pull-down menu (in the List View screen or a Record screen), click on the Category pull-down menu and select the last selection, **Edit Categories**. This will open the List Editor screen with the Category list selected.

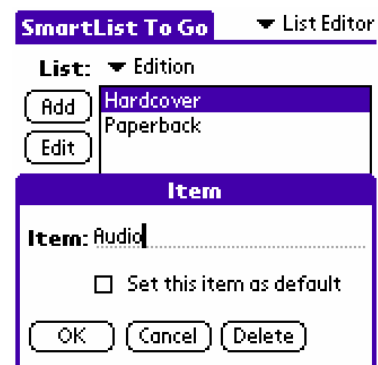


Configuring the Category List

Configuring the Category list in the List Editor screen includes adding, removing, modifying and sorting categories.

To add a category, follow these steps:

1. Open the List pull-down menu at the top of the screen and select **Category**.
2. Click on the **Add** button to add the first category to the list. This will open the Item screen.
3. Enter the text for the first category. Click **OK** on the Item screen to complete the addition or **Cancel** to close the Item screen without adding the category.
4. Click **Done** on the List Editor screen to return to the previous screen



By default, the Category menu in a new record will remain **Unfiled** until you specifically select a value from the list.

To automatically assign all new records to a default category, follow these steps:

1. Open the List pull-down menu at the top of the screen and select **Category**.
2. Select the category you wish to set as the default and click on the **Edit** button to modify it. This will open the Item screen.
3. Check the box labeled Set this item as default. Click **OK** on the Item screen to complete the modification or **Cancel** to close the Item screen without setting the default category.
4. Click **Done** on the List Editor screen to return to the previous screen.

You may override this default at any time by manually assigning the record to any category as described in the section on Assigning Categories below.

To modify a category, follow these steps:

1. Open the List pull-down menu at the top of the screen and select **Category**.
2. Select the category you wish to edit and click on the **Edit** button to modify it. This will open the Item screen.
3. Revise the text for the category. Click **OK** on the Item screen to complete the modification or **Cancel** to close the Item screen without changing the category.
4. Click **Done** on the List Editor screen to return to the previous screen.

To remove a category, follow these steps:

1. Open the List pull-down menu at the top of the screen and select **Category**.
2. Select the category you wish to edit and click on the **Edit** button to open the Item screen.
3. Click **Delete** on the Item screen to remove the category or **Cancel** to close the Item screen without removing the category.
4. Click **Done** on the List Editor screen to return to the previous screen.

To sort a category list, follow these steps:

1. Open the List pull-down menu at the top of the screen and select **Category**.
2. Sort the list by selecting individual categories and moving them up or down in the list. To do this, select the category you wish to move and click on the **Up** and **Down** buttons as required.

To quickly sort the entire list, click on the **Sort** button. This will put the categories into alphabetical order.

3. Click **Done** on the List Editor screen to return to the previous screen.

5.1.2 Assigning Records to a Category

Assign a record to a category from the Record Screen.

To assign a record to a category, open the Category pull-down menu at the top right corner of the screen. Select the desired category to close the pull-down menu and display the record's new category.

5.1.3 Displaying by Category

Display records according to category in the List View screen.

To display records according to category, open the Category pull-down menu at the top right corner of the screen. Select the desired category to close the pull-down menu and display the record's new category. This will create a List View that includes only those records which belong to the category you selected.

To display records that do not belong to a category, select **Unfiled** in the Category pull-down menu.

To display all records, regardless of category, select **All** in the Category pull-down menu.

5.2 The Expression Field

The Expression, Advanced Expression and Conditional Expression field types depend upon mathematical or boolean expressions for their values. These expressions may include the values in other fields in combination with constant values.

The **Expression** field type determines its value according to a mathematical expression you design. Unlike other fields that receive their values through user input, the SmartList To Go handheld application calculates the result of the expression and automatically displays the answer in the field.

The Reading tab in the SmartList MyBooks contains two examples of Expression fields, "Pg left" and "%read". The expression field "Pg left" subtracts the value in the field "Last Pg" from the value in the field "Pages" and displays the result. You cannot modify this result directly, but if a value in "Last Pg" or "Pages" changes, the SmartList To Go handheld app will recalculate and display the new value.

Similarly, the "%read" field uses the values in "Pages" and "Pg Left" to calculate the percentage of pages read so far.



Note: The SmartList To Go handheld application only recalculates expression fields to account for changes when certain triggering events occur, such as when you click on the expression field on the Record form. Simply changing the value of one of the fields used in the expression will NOT necessarily cause the SmartList To Go app to recalculate. Therefore, if you have made any changes at all to the fields, always make sure that the SmartList To Go handheld app recalculates expressions. To do this, follow the instructions for Manually Recalculating Expressions below.

The **Conditional Expression** field type determines its value according to a true or false question. When you create a conditional expression field, you assign two possible values for the field and a boolean expression to determine which value to use. A **boolean expression** is simply an equation that compares two values and returns a true or false answer. If the comparison is correct, the boolean expression is true, and the field returns the first value. If the comparison is not correct, the boolean expression is false, and the field returns the second value.

Let's clarify that explanation a little with an example. Suppose that you wanted to create a field in MyBooks that would indicate whether or not you were more than halfway finished with a book. To do this, you could create a conditional expression field called "Half Progress" with the two possible values "Less than half way" and "More than half way" and the boolean expression:

Last Page < Pages / 2

When you enter a values in the fields "Last Page" and "Pages," SmartList To Go will divide the total number of pages by two to determine the halfway point in the book and then compare that number with the last page you have read.


The expression above states that the last page you read is less than the halfway point of the book. If that statement is true, the field will return the response "Less than half way;" if that statement is false, the field will return the response "More than half way."

In This Section . . .

- 5.2.1 Creating an Expression Field
- 5.2.2 The Expression Builder Screen

5.2.1 Creating an Expression Field

To create an expression field, you will follow the same basic procedure as you will when creating any kind of field: first open the Field Editor Screen, second select the New button, third enter a name for the new field and choose Expression as the type. Tap the Done button to return to the Field Editor. The steps below describe this process in detail as it applies to an Expression field. Begin this process in the Field Editor screen:

1. Tap the New button to open the Single Field Editor.
2. Enter the name of the field in the column labeled "Name."
3. Choose **Expression** as the Type.
4. Click on the Field Properties icon  to open the Expression Builder screen. Design the expression in this screen and click **OK** to finish. The next section describes how to build expressions in this screen.
5. Open the pull-down menu in the upper right corner of the screen and select **Form Designer** to open the Form Designer screen.
6. Add the field to the form as described in Chapter 4.



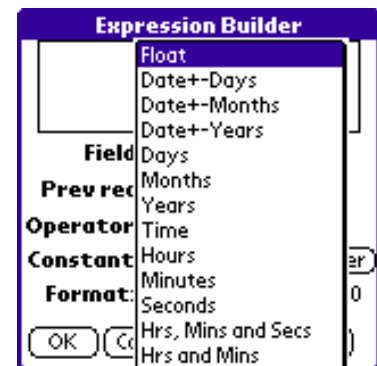
The Expression field is now created.

5.2.2 The Expression Builder Screen

As described in the section above, you will use the Expression Builder screen to design expressions for Expression type fields. Open this screen as described above.

To create an expression in the Expression Builder screen, follow these steps:

1. Select the format for the result using the Format pull-down menu at the bottom of the screen. The table below lists the available formats. Note that certain formats are limited in the types of data they can accept as factors.



Expression Field Type Result Formats and their Requirements		
Type	Description	Factor Requirements
Float	Float format results are numbers. They may be integers, or they may include up to three digits after the decimal.	All factors must be numerical. The operator may be plus, minus, multiply or divide.
Date+-Days	The Date+-Days format adds or subtracts days from a Date type field value and returns the sum or difference in a Mon/Day/Yr format. For example, the expression 2/17/01 + 2 would evaluate to 2/19/01.	The first factor must be a Date type field value, the operator must be plus or minus and the second factor must be numerical field type value.
Date+-Months	The Date+-Months format adds or subtracts months from a Date type field value and returns the sum or difference in a Mon/Day/Yr format. For example, the expression 2/17/01 + 2 would evaluate to 4/17/01.	The first factor must be a Date type field value, the operator must be plus or minus and the second factor must be numerical field type value.
Date+-Years	The Date+-Years format adds or subtracts years from a Date type field value and returns the sum or difference in a Mon/Day/Yr format. For example, the expression 2/17/01 + 2 would evaluate to 2/17/03.	The first factor must be a Date type field value, the operator must be plus or minus and the second factor must be numerical field type value.
Days	The Days format adds or subtracts two Date type field values and returns the sum or difference in days.	Both factors must be Date type field values, and the operator must be a plus or minus.
Months	The Months format adds or subtracts two Date type field values and returns the sum or difference in months.	Both factors must be Date type field values, and the operator must be a plus or minus.
Years	The Years format adds or subtracts two Date type field values and returns the sum or difference in years.	Both factors must be Date type field values, and the operator must be a plus or minus.
Time	The Time format adds or subtracts seconds from a Time field type value and returns the sum or difference in Hrs:Mins:Secs format. For example, the expression 3:30:00 + 2 would evaluate to 3:30:02.	The first factor must be a Time type field value, the seconds factor must be a numerical field type value, and the operator must be plus or minus.
Hours	The Hours format adds or subtracts two Time field type values and returns the sum or difference in hours.	Both factors must be Time type field values, and the operator must be a plus or minus.
Minutes	The Minutes format adds or subtracts two Time field type values and returns the sum or difference in minutes.	Both factors must be Time type field values, and the operator must be a plus or minus.
Seconds	The Seconds format adds or subtracts two Time field type values and returns the sum or difference in seconds.	Both factors must be Time type field values, and the operator must be a plus or minus.
Hrs, Mins and Secs	The Hrs, Mins and Secs format adds or subtracts two Time type field values and returns the sum or difference in an Hrs:Mins:Secs format.	Both factors must be Time type field values, and the operator must be a plus or minus.
Hrs and Mins	The Hrs, Mins and Secs format adds or subtracts two Time type field values and returns the sum or difference in an Hrs:Mins:Secs format.	Both factors must be Time type field values, and the operator must be a plus or minus.

2. Construct the expression in the box at the top of the screen using the following controls.

Expressions will evaluate sequentially. For example, if you create the expression:

$$1 + 3 / 2 - 1$$

The order of the operations will be:

- $1 + 3 = 4$
- $4 \text{ (the result from the first portion of the equation)} / 2 = 2$
- $2 \text{ (the result from the second portion of the equation)} - 1 = 1$

Build the expression sequentially as well, starting from left and proceeding to the right as though you were writing the expression on paper.

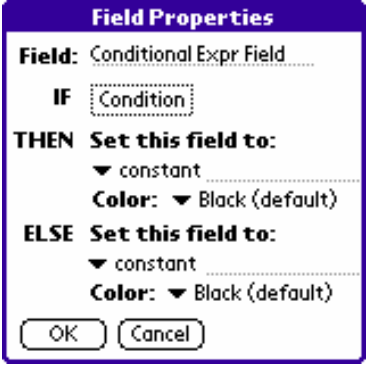
Expression Builder Screen Controls	
Field	<p>Add the value of another field in this record to the expression. When SmartList To Go calculates the value of this expression, it will use whatever value appears in this field to do so. You can use almost any field type as long as it exists in the same SmartList.</p> <p>To add a field to the expression using the Field Control, click on the down arrow to open a pull-down menu of all the eligible fields in the SmartList. Select the desired field to close the menu and add the field to the expression.</p>
Prev rec	<p>Add the value of a field from the previous record to the expression. When SmartList To Go calculates the value of this expression, it will use whatever value appears in this field in the previous record to do so. Almost any field type is permissible.</p> <p>To add a field to the expression using the Prev rec Control, click on the down arrow to open a pull-down menu of all the eligible fields in the SmartList. Select the desired field to close the menu and add the field to the expression.</p>
Operator	<p>Add a +, -, * or / operator between two factors to add, subtract, multiply or divide them respectively. Whether a constant or a field value, ALL factors MUST be separated by an operator for the expression to be valid.</p> <p>To add an operator to the expression using the Operator buttons, click on the desired button.</p>
Constant	<p>Add a constant numerical value to the expression.</p> <p>To add a constant to the expression using the Constant Control, first enter the constant value on the line provided. Then click Enter to add the constant value to the expression. This will clear the line in this control to make it ready for the next value.</p>
Back	Click this button to delete the last item (field, constant or operator) you added to the expression.
Clear	Click this button to clear the entire expression.

3. Select the number of digits to include after the decimal in the **Decs** menu.
4. Click **OK** to complete the expression design, or click **Cancel** to close the Expression Builder screen without applying the changes to the expression you have just made. The Expression will not produce a value of any kind if the Expression Builder screen is empty.

The expression is now created.

5.3 The Conditional Expression Field

A Conditional Expression field contains a value automatically determined depending on the values entered in one or more other fields. When you design a Conditional Expression field, you make a statement about a value in another field and two possible values. In a record in which the statement you have designed is true, the Conditional Expression field will automatically contain the first value; in a record in which the statement you have designed is false, the Conditional Expression field will automatically contain the second value.



The image shows a 'Field Properties' dialog box. The 'Field' is named 'Conditional Expr Field'. Under the 'IF' section, there is a text box containing 'Condition'. The 'THEN' section is titled 'Set this field to:' and has a dropdown menu set to 'constant'. Below this, the 'Color' is set to 'Black (default)'. The 'ELSE' section is also titled 'Set this field to:' with a dropdown menu set to 'constant', and its 'Color' is also 'Black (default)'. At the bottom are 'OK' and 'Cancel' buttons.

Your primary interfaces for designing a Conditional Expression type field are the Conditional Expression Field Properties screen and the Condition screen.

In This Section . . .

- 5.3.1 The Parts of a Conditional Expression
- 5.3.2 Creating a Conditional Expression
- 5.3.3 The Conditional Expression Properties Screen

5.3.1 The Parts of a Conditional Expression

Take a look at the Conditional Expression Properties Field to the right. You will use this screen to define the Conditional Expression type fields.

In this screen, you will see that the properties of a Conditional Expression field have three main parts that define how the field will behave. These parts are, the IF Condition, the THEN Response and the ELSE Response.

Stated simply, the **IF Condition** is a statement that is either true or false. The **THEN Response** and **ELSE Response** indicate two possible values for the Conditional Expression type field. If the IF Condition is true, the field uses the value indicated in the THEN Response; if the IF Condition is false, the field uses the value indicated in the ELSE Response.

Essentially, you are creating a field with two possible values. You are saying, "If this statement is true, use the first value; if it is false, use the second value."

The IF Condition

An IF Condition is in the form of mathematical expression that compares two values. For example, in the expression:

$$2 > 3$$

the IF Condition is stating that two is greater than three. In this case, the IF Condition will reply that the statement is false, because two is not greater than three.

Conditional expressions can compare values in a number of different ways. In the expression above, the conditional expression states that one value is greater than or less than another, but you can also say that two values are equal or not equal, that one value contains the other value or does not, or that one value starts with the other value. In every case, however, there are only two possible responses from the IF Condition: true or false.

Another example of an IF Condition is:

Field 1 starts m

In this example, Field 1 is a Text type field. If the text you entered in Field 1 in this record starts with the letter "m," then this IF Condition will reply that the statement is true; if it does not, the IF Condition will reply that the statement is false. This is significant; unlike the first example, which is always false, you have made the IF Condition's response dependent upon the value entered in another field. As a result, some records will evaluate this condition and find that it is true, and others will evaluate this condition and find that it is false.


The THEN and ELSE Responses

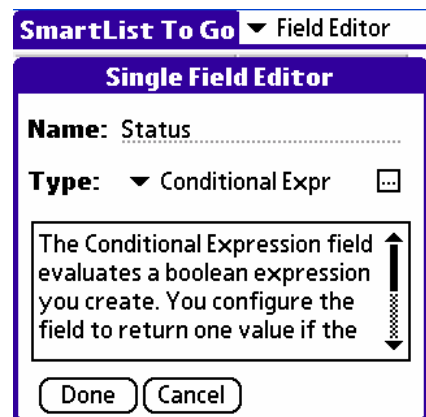
The **THEN Response** and **ELSE Response** each indicate a value to place in the conditional expression field. The state of the IF Condition determines which Response is used. If the Condition is true, the value described by the THEN Response is used.

THEN and ELSE Response can be constant values, or they can refer to other fields. For example, say that you were creating a SmartList which contained two fields that accepted values describing an item's price from competing vendors. You could create a third field, a Conditional Expression, to always determine which price was cheapest. The field would test to see if the first price was less than the second. If it was, the field would display the first price; if it was not, the field would display the second price, as that one must then be the lower of the two.

5.3.2 Creating a Conditional Expression

To create a conditional expression field, you will follow the same basic procedure as you will creating any kind of field: first open the Field Editor Screen, second select the New button, third enter a name for the new field and choose Conditional Expression as the type. Tap the Done button to return to the Field Editor. The steps below describe this process in detail as it applies to an Conditional Expression field. Begin this process in the Field Editor screen:

1. Tap the New button to open the Single Field Editor.
2. Enter the name of the field in the column labeled "Name."
3. Choose **Conditional Expr** as the Type.
4. Click on the Field Properties icon  to open the Conditional Expressions Field Properties screen. Design the expression in this screen and click **OK** to finish. The next section describes how to build conditional expressions in this screen.



5. Open the pull-down menu in the upper right corner of the screen and select **Form Designer** to open the Form Designer screen.
6. Add the field to the form as described in Chapter 4.

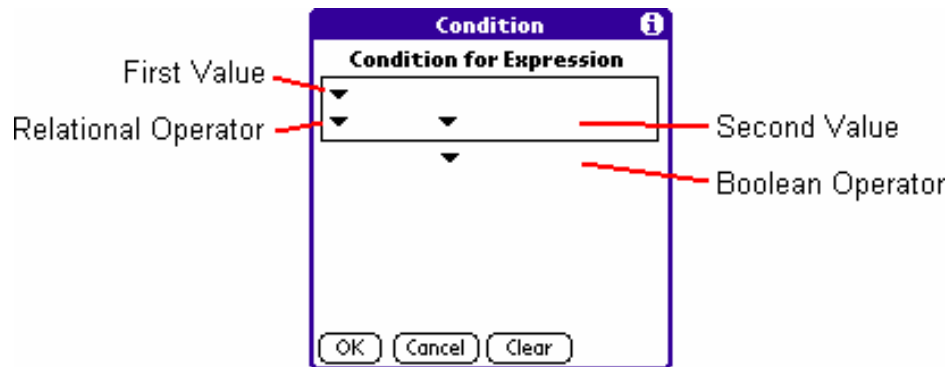
The Conditional Expression field is now created.

5.3.3 The Conditional Expression Properties Screen

As described in the section above, you will use the Conditional Expression Field Properties screen to design the expressions for Conditional Expression type fields. Open this screen as described above.

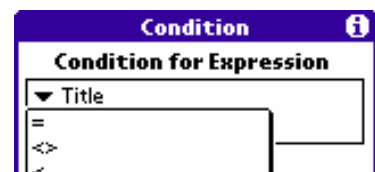
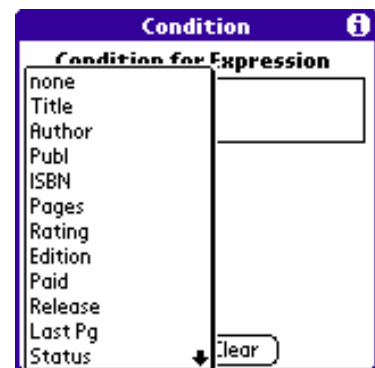
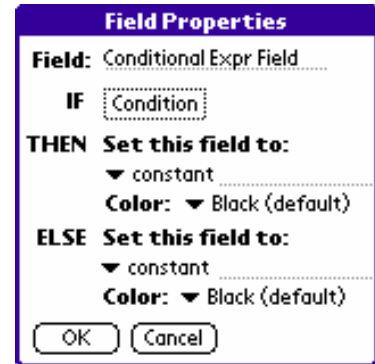
To create a conditional expression in the Conditional Expression Properties screen, follow these steps:

1. Click on the Condition button to open the Condition screen. You will use this screen to design the IF Condition.



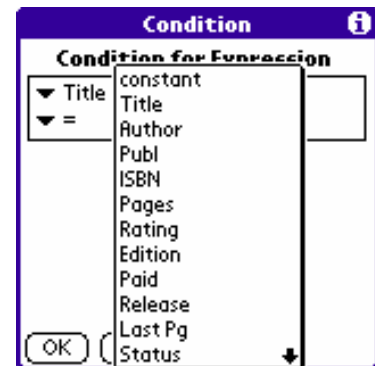
A Conditional Expression Consists of three basic components: the First Value, the Second Value, and the Relational Operator that compares those values. You will design these three components in this screen.

2. Click on the First Value down arrow to open a pull-down menu of possible values. By selecting one of the other fields in this manner, you enter the value in that field as the First Value in the IF Condition.
3. Click on the Relational Operator down arrow to open a pull-down menu of possible relational operators. These will define how to compare the two values. The operators available include:



- =
- <> (not equal)
- <
- >
- <=
- >=
- contains
- Contains (case sensitive)
- starts
- Starts (case sensitive)

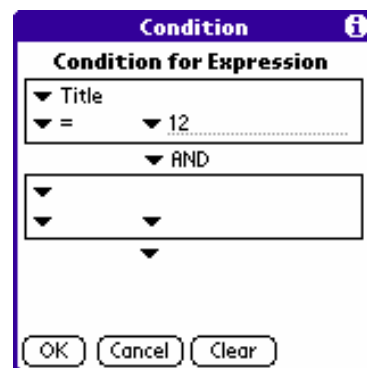
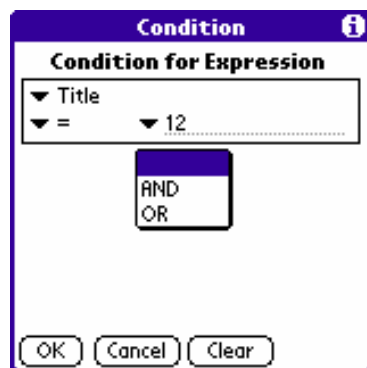
- Click on the Second Value down arrow to open a pull-down menu of possible values. By selecting one of the other fields in this manner, you enter the value in that field as the First Value in the IF Condition. You may also select **constant** to enter the Second Value directly, as shown in the examples to the right.



- Click on the Logical Operator down arrow to open a pull-down menu containing the choices AND and OR. If you select one of the choices, the screen will display inputs for a second condition. Repeat steps 2, 3 and 4 to create the second condition.

If you select AND, the IF Condition will return a value of true only when both the conditions you defined are true. If you select OR, the IF Conditions will return a value of false when at least one of the conditions is true.

Use these selections to define up to three separate conditions to test in the Conditional Expression Field. If you select three conditions, they will evaluate in order.



- Click **OK** to return to the Conditional Expression Properties Field and apply the IF Condition design or click Cancel to close the Condition screen without applying the IF Condition design.

7. Define the value to apply to the field when the IF Condition is true in the THEN Response portion of the screen. Click on the down arrow in to open a pull-down menu including the other field names and **constant**.

If you select a field name, this Conditional Expression field will return the value in the field you selected when the IF Condition is true. If you select **constant**, you can enter a value directly that the Conditional expression field will return when the IF Condition is true.

The screenshot shows a dialog box titled "Field Properties". Inside, there is a section for "Field:" with the text "Conditional Expr Field". Below this is a section for "IF" with a "Condition:" label. To the right of the "IF" section is a "THEN" section with a "Value:" label. A pull-down menu is open, showing a list of options: "constant", "Title", "Author", "Publ", "ISBN", "Pages", "Rating", "Edition", "Paid", and "Release". The "THEN" section has a "Value:" label and a text input field containing "(default)".

8. Repeat Step 7 in the ELSE Response portion of the screen to define the value to apply to the field when the IF Condition is false. Click on the down arrow in to open a pull-down menu including the other field names and **constant**.

If you select a field name, this Conditional Expression field will return the value in the field you selected when the IF Condition is false. If you select **constant**, you can enter a value directly that the Conditional expression field will return when the IF Condition is false.

9. Click **OK** to close the Conditional Expression Properties screen and apply the properties you have defined, or click **Cancel** to close the screen without applying the properties.

10. Add the newly defined Conditional Expression field to the form as described in Chapter 4.

The Conditional Expression type field is now created.

5.4 The Advanced Expression Field

The **Advanced Expression** field type is a field whose value depends upon an expression. Unlike other fields that receive their values through user input, the SmartList To Go handheld application calculates the result of the expression and automatically displays the answer in the field.

There are three possible types of expressions you can use: numerical, boolean and string (text).

- A **numerical** expression is a mathematical phrase that evaluates to a constant value. In SmartList To Go, such an expression can be as simple as two plus two, but more complex numerical expressions may incorporate advanced operations like square root, parenthesis, values from other fields in the same or the previous record, variables like pi and constants like the date the record was created. Numerical expressions include expressions that manipulate dates and times as well, including a variety of functions with which to do so.
- A **boolean** expression compares the values of two mathematical phrases and evaluates to either true or false. For example, $1 + 1 = 2$ is a boolean expression that is true, and $1 + 1 < 2$ is a boolean expression that is false. These expressions can vary in complexity as widely as a numerical expression. When you use an Advanced Expression field to create a boolean expression, you will indicate two values to go with the expression. If the boolean expression is true, the field will display the first value; if the expression is false, the field will display the second value.
- A **string (text)** expression manipulates text instead of numerical values. For example, "birth" + "day" is a string (text) expression that would evaluate to "birthday". String (text) expressions may incorporate values from other fields or other records, parenthesis, variables and constants.

The Advanced Expression is a new field type for the SmartList To Go handheld app. It encompasses the capabilities of both the Expression and Conditional Expression field types that were discussed in the previous two sections and more. However, the SmartList To Go handheld application will continue to use the Expression and Conditional Expression field types so that SmartLists created in previous version can be viewed and modified without alteration.



Note: The SmartList To Go handheld application only recalculates expression fields to account for changes when certain triggering events occur, such as when you click on the expression field on the Record form or you resort the records in the List View screen. Simply changing the value of one of the fields used in the expression will NOT necessarily cause the SmartList To Go handheld app to recalculate. Therefore, if you have made any changes at all to the fields, always make sure that the SmartList To Go handheld application recalculates expressions. To do this, follow the instructions for Manually Recalculating Expressions below.


In This Section . . .

- **5.4.1 Creating an Advanced Expression Field**
- **5.4.2 The Advanced Expression Builder Screen**
 - **5.4.2.1 Expression Design Inputs and Controls**
 - **5.4.2.2 Advanced Expression Result Formats**

- **5.4.2.3 Notes on Advanced Expression Syntax**

5.4.1 Creating an Advanced Expression Field

To create an **Advanced Expression field**, you will follow the same basic procedure as you will creating any kind of field: first open the Field Editor Screen, second select the New button, third enter a name for the new field and choose **Adv Expression** as the type. Tap the Done button to return to the Field Editor. The steps below describe this process in detail as it applies to an Advanced Expression field. Begin this process in the Field Editor screen:

1. Tap the New button to open the Single Field Editor.
2. Enter the name of the field in the column labeled "Name."
3. Choose **Adv Expression** as the Type.
4. Click on the Field Properties icon  to open the Expression Builder screen. Design the expression in this screen and click **OK** to finish. The next section describes how to build expressions in this screen.
5. Open the pull-down menu in the upper right corner of the screen and select **Form Designer** to open the Form Designer screen.
6. Add the field to the form as described in Chapter 4.

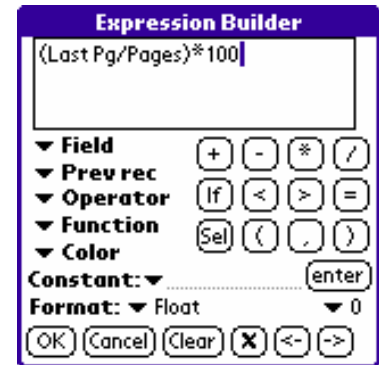
5.4.2 The Advanced Expression Builder Screen

As described in the section above, you will use the Advanced Expression Builder screen to design expressions for Advanced Expression type fields.

To create an expression in the **Expression Builder screen**, begin in the Advanced Expression Builder screen and perform the tasks below. Of course, the exact steps you take will vary depending upon what you need the expression to do, but the process will always include these three general tasks:



1. Create the expression in the box in the upper portion of the Advanced Expression Builder screen using the inputs and controls in the lower half of the screen. Section 5.4.2.1 Expression Design Inputs and Controls below describes these controls in detail.
2. Select a format for displaying the results. The format helps you control things like how many digits come after the decimal point in a numerical value, whether a time value displays the seconds or whether the result of the expression displays as text or numerical, just to give a few examples. Section 5.4.2.2 Advanced Expression Result Formats below describes the various formats in detail.
3. Click **OK** to apply your changes and return to the Field Designer screen or click **Cancel** to close the Advanced Expression Builder screen without applying your changes.



5.4.2.1 Expression Design Inputs and Controls

The Advanced Expression Builder screen provides a wide variety of controls for creating expressions. An expression is a series of elements separated by operators that evaluates to a single value. These expressions may be simple, like $2 + 2$, or they may be very complex, incorporating up to 48 constants, variables, fields and advanced mathematical, date or time functions as elements with advanced mathematical and logical operators to manipulate them.

To add elements to an expression, open the appropriate pull-down menu on the left side of the screen and select the element. These menus divide the elements into the following groups:

- **Fields** - This group includes fields from the current record. Open this menu and select the name of the field you wish to add to the expression. When the SmartList To Go handheld app evaluates the expression, it will use the value stored in that field in the current record.
- **Prev rec** - This group includes fields from the previous record. Open this menu and select the name of the field you wish to add to the expression. When the SmartList To Go app evaluates the expression, it will use the value stored in that field in the previous record.
- **Operator** - This group includes logical operators like **AND** and **NOT** and the mathematical operator **%**, or modulo. Other operators in addition to those in this menu - mathematical operators like **plus** and **minus** and comparative operators like **greater than** and **equals** and string (text) operators like **concatenate** - are all available using the buttons to the right of these menus.
- **Function** - This group provides advance functionality for calculating square roots and rounding values and for working with date and time values.
- **Color** - This group allows you to assign color to specific value options when using the **If** or **Selection** functions. For example, you might create a conditional expression to compare a bill's due date with the current date; using the color pull-down menu, you could ensure that the due date of overdue bills appeared in red. Note that colors appear as shades of gray on monochrome display screens.

In addition, the twelve buttons on the right side of the screen provide quick access to the ten most commonly used operators and the commonly used functions If and Selection.

Operators

The following table describes the available operators in order of precedence from highest to lowest. Operators with higher precedence are evaluated before operators with lower precedence. For example, the SmartList To Go handheld application evaluates the expression

$$2 + 3 * 4$$

by first multiplying three times four, then adding two because multiplication has a higher precedence than addition. Thus, this expression evaluates to fourteen.


The buttons on the right side of the screen provide access to the ten most commonly used operators; use the Operators pull-down menu for all others.

! (NOT)	The logical operator NOT appears on the Operator pull-down menu as "NOT" and in the Expression view area as an "!" exclamation point. Use this operator for conditional expressions.
**	This operator raises the base number that precedes it to the power indicated by the integer that follows. For example, 2**3 indicates 2 ³ , or two to the third power.
*, / and %	The Multiply, Divide and Modulo operators respectively. Modulo, or simply Mod, divides the first element by the second elements and returns the remainder of that division, so that 5 % 2 evaluates to 1, since 5 divided by 2 is 2 with a remainder of 1.
+ and -	The Plus and Minus operators. Use the Plus operator for addition in mathematical phrases, as in 2 + 2, which evaluates to 4, or for Concatenation in text string (text) phrases, such as "birth" + "day", which evaluates to 'birthday'. Use the Minus operator for subtraction in mathematical phrases, as in 6 - 4.
<, >, <= and >=	The Less Than, Greater Than, Less Than or Equal To and Greater Than or Equal to operators. Create the Less Than or Equal To and Greater Than or Equal to operators by combining Less Than or Greater Than with Equals.
= and <>	The Equals and Does Not Equal operators. Create the Does Not Equal operator by combining Less Than and Greater Than.
& (AND)	The logical operator AND appears on the Operator pull-down menu as "AND" and in the Expression view area as an "&" ampersand. Use this operator for conditional expressions.
^ (NOR)	The logical operator NOR appears on the Operator pull-down menu as "NOR" and in the Expression view area as a "^" caret. Use this operator for conditional expressions.
 (OR)	The logical operator OR appears on the Operator pull-down menu as "OR" and in the Expression view area as a " " vertical line. Use this operator for conditional expressions.
(and)	The Open Parens and Close Parens operators serve two very different purposes. First, these operators indicate the order of operation when that order does not follow the normal order of precedence. Consider, for example, the equation 2 + 3 * 4 Following the order of precedence shown in this table, SmartList To Go would first multiply three and four, then add two for a total of fourteen. However, by adding parenthesis as follows (2 + 3) * 4 you indicate to SmartList To Go that it should add two and three first, then multiply the result by four for a total of twenty. Second, these operators also enclose the elements of a function. For example, Sel(numberGrade / 10 - 4, 'F', 'D', 'C', 'B', 'A') Functions insert the Open parens automatically.
,	The Comma operator. This operator separates items in a function, as in the example above.

Numerical Functions

The following table describes the numerical functions available in the Functions pull-down menu as well as the If and Selection functions, which are included in the buttons on the right side of the screen.

If	<p>The Conditional or If function has three arguments: the Condition, True Value and False Value. Enter Conditional functions using the format :</p> <p style="text-align: center;">If([Condition], [Then Result], [Else Result])</p> <p>The Condition is a boolean expression and is always the first argument. Boolean expressions compare two values and are either true or false. If the Condition evaluates to true, the function evaluates to the True Value; if the Condition is false, the function evaluates to the False Value.</p> <p>For an example, consider the following expression:</p> <p style="text-align: center;">If(customerAge >= 21, 'Grant ', 'Deny ') + 'Entry'</p> <p>This expression will compare the value entered in the field called customerAge to the literal value 21. If the value in customerAge is greater than or equal to 21, the Condition is true, and the function will return the True Value 'Grant Entry'. If not, the Condition is false, and the function will return the False Value 'Deny Entry '.</p> <p>Notice that the three items in the Conditional Function are separated by commas, and the entire group is enclosed by parenthesis.</p>
Sel	<p>The Selection function has three or more arguments: The Selector and at least 2 Values. There is no direct limit to the number of possible values you may use, but the total length of the expression cannot exceed 48 elements.</p> <p>Write Selection functions using the format:</p> <p style="text-align: center;">Sel([Selector], [1st Value], [2nd Value], . . .)</p> <p>The Selector is a mathematical expression that evaluates to an integer and is always the first argument. That integer determines which of the values that follow the functions returns. If the Selector evaluates to 1, the function will evaluate to the 1st Value; if the Selector evaluates to 2, the function will evaluate to the 2nd Value, and so on. If the Selector evaluates to an integer that does not have a corresponding Value, the function will evaluate to zero. For an example, consider the following expression:</p> <p style="text-align: center;">Sel(numberGrade / 10 - 4, 'F', 'D', 'C', 'B', 'A')</p> <p>A teacher could use this expression to translate number grades into letter grades. Assuming that the value entered in the field called numberGrade is a number from 0 to 99, the Selector would divide that grade by ten and subtract 4 from the result. Number grades from 50 to 59 would result in an Selector value of 1, so the function would return the value 'F'. Number grades from 60 to 69 would result in an Selector value of 2, so the function would return the value 'D'. And so on, up to value from 90 to 99 that would result in a Selector value of 5, and the function would return the value 'A'.</p> <p>Notice that the items in the Selection Function are separated by commas, and the entire group is enclosed by parenthesis.</p>

	 Note: If the Selector evaluates to an integer that does not have a corresponding Value, the function will evaluate to the text string (text) "Error". If you have selected a Format other than text in which to display this field's results, however, the string (text) "Error" will appear as a zero.
Sqrt	<p>The Square Root function has one argument and calculates its square root. For example,</p> $\text{Sqrt}(\text{lengthA} * \text{lengthA} + \text{lengthB} * \text{lengthB})$ <p>will first evaluate the expression inside the function, then return the square root of the result.</p>
Round	<p>The Round function has one argument that is a number with a decimal point or an expression that evaluates to a number with a decimal point. The Round function will convert the number to the nearest integer.</p> <p>For example, if the argument evaluated to 3.14, the Round function would return the value 3. If the argument evaluated to 6.91, the Round function would return the value 7.</p>
Ceil	<p>The Ceiling function has one argument that is a number with a decimal point or an expression that evaluates to a number with a decimal point. The Ceiling function will round the number up to the next integer.</p> <p>For example, if the argument evaluated to 4.15, the Ceiling function would return the value 5.</p>
Floor	<p>The Floor function has one argument that is a number with a decimal point or an expression that evaluates to a number with a decimal point. The Floor function will round the number down to the last integer.</p> <p>For example, if the argument evaluated to 2.96, the Floor function would return the value 2.</p>
Abs	<p>The Absolute Value function has one numerical argument. If the argument evaluates to a positive number, the function will return the positive number, but if the argument is negative, the function will convert the result to a positive number.</p>
Sign	<p>The Sign function has one numerical argument. If the argument evaluates to a positive number, the function will return the value 1. If the argument evaluates to a negative number, the function will return the value -1. If the argument evaluates to zero, the function will return zero.</p>

Date and Time Functions

The following table describes the functions available in the Functions pull-down menu that relate to date and time values.

Date	<p>The Date function has three numerical arguments: month, day and year. Write the Date function in the following format:</p> <p style="text-align: center;">Date([Month], [Day], [Year])</p>
Time	<p>The Time function has three numerical arguments: hour, minute and second. Write the Time function in the following format:</p> <p style="text-align: center;">Time([Hour], [Minute], [Second])</p>
Year, Month, and Day	<p>The Year, Month and Day functions all operate in a similar way. These functions have one argument in Date format, and they extract and return the corresponding component of that Date.</p> <p>For example,</p> <p style="text-align: center;">Year(Birthday)</p> <p>in which Birthday is another field that contains the Date type value 7/31/1975, would return the value 1975.</p>
Hour, Minute, and Second	<p>The Hour, Minute and Second functions all operate in a similar way. These functions have one argument in Time format, and they extract and return the corresponding component of that Time.</p> <p>For example,</p> <p style="text-align: center;">Minute(Start Time)</p> <p>in which Start Time is another field that contains the Time type value 4:47:02 pm, would return the value 47.</p>
Weekday	<p>The Weekday function has one argument in Date format and returns an integer from one to seven that corresponds to the day of the week the date describes. A one indicates Sunday, a 2 indicates Monday and so on up to seven, which indicates Saturday.</p> <p>For example, if the date 12/22/1981 fell on a Thursday, the expression</p> <p style="text-align: center;">Weekday(12/22/1981)</p> <p>would return the value 5.</p>

5.4.2.2 Advanced Expression Result Formats

The second phase of creating an advanced expression is determining a format for expressing the result. Results may be displayed as numbers, text, a date or a time in a variety of combinations of hours, minutes and seconds.

Your choice of format will depend upon the type expression you create, but the format will have no effect on how the expression calculates results. Formats affect the display only. For example, consider the expression

`7 + 'eleven'`

The SmartList To Go handheld application will evaluate this expression numerically because the numerical component comes first. It will begin with a value of seven and add zero because 'eleven' is not a number, and has no numerical value. So the result of this expression is seven.

You might be tempted to think that setting the result format to **string (text)** (text) would force the SmartList To Go handheld application to treat both the elements as string (text)s, leading to a result of "7eleven". This, however, is not the case.

The result of this expression remains seven regardless of the display format you choose. If you select a numerical result format, the field will display the answer seven as a number. This number can then be used as an input in other expressions that will treat it as a number. If you select a text string (text) result format, the field will display the answer seven as text. The character seven can then be used in other expressions that will treat it as a text character.

The table below provides a complete description of the available result formats.

Float	<p>The Float result format displays the result of an expression as a numerical value.</p> <p>For example, an expression that added 2 and 2 would display the result 4, and the SmartList To Go handheld application would treat that result as a number if you then used that value in a subsequent expression.</p> <p>However, an expression that concatenated the text string (text)s 'Ronald ' and 'Reagan' would display the result, 'Ronald Reagan' as zero, since this text string (text) has no numerical value.</p> <p>The Float result format, when selected, is accompanied by a second control that enabled you to select how many digits to display after the decimal. You may choose from zero digits (this will display integers only) to three digits.</p> <p>For example, an expression that added 2 and 2 in which you had indicated to display two digits after the decimal would display the result 4.00.</p> <p>The Float result format, when necessary, will truncate results. It does not round. Thus, an expression that added 1.2 and 1.4 in which you indicated no digits after the decimal would display the result 2.</p> <p>To round results instead of truncating, use the Round function described in the previous section.</p>
Date	<p>The Date result format displays the result of an expression as three separate numerical values divided by forward slashes as in the example</p> <p style="text-align: center;">7/31/75</p> <p>The first value is an integer that indicates the number of the month, the second value indicates the day of the month and the third value indicates the year.</p> <p>It is important to use the Date result format to display Date values only. Numerical, text or time values will result in no display or values that won't make any sense.</p>
Time	<p>The Time result format displays the result of an expression as three separate numerical values divided by a colon with the addition of the letters "am" or "pm", as in the example</p> <p style="text-align: center;">12:54:08 pm</p> <p>This first value is an integer that indicates the hour, the second indicates the minutes and the third indicates seconds.</p> <p>It is important to use the Time result format to display Time values only. Numerical, date or text values will result in no display or values that won't make any sense.</p>
Hours	<p>The Hours result format displays only the hours component of a Time value. For example, an expression that evaluates to the Time value 12:54:08 pm in which the Hours result format is selected will display the integer 12.</p>
Minutes	<p>The Minutes result format displays only the minutes component of a Time value. For example, an expression that evaluates to the Time value 12:54:08 pm in which the Minutes result format is selected will display the integer 54.</p>

Seconds	The Seconds result format displays only the seconds component of a Time value. For example, an expression that evaluates to the Time value 12:54:08 pm in which the Seconds result format is selected will display the integer 8.
Hrs, Mins and Secs	<p>The Hrs, Mins and Secs format displays the result of an expression as three separate numerical values divided by a colon, as in the example</p> <p style="text-align: center;">12:54:08</p> <p>As in the Time result format, the first integer indicates the hour, the second the minutes and the third the seconds. Unlike the Time result format, however, the "am" or "pm" indicator does not appear.</p>
Hrs and Mins	<p>The Hrs and Mins format displays the result of an expression as two separate numerical values divided by a colon, as in the example</p> <p style="text-align: center;">12:54</p> <p>The first integer indicates the hour and the second indicates the minutes.</p>
Text	<p>The Text format displays the results of an expression as literal text. Unlike the Date format, which can only display Date type values, or the Time format, which can only display Time type values, the Text format can display any kind of value, whether it is text, numerical, a date or a time.</p> <p>However, any value you display as text will be considered text from that point on. For example, if you created Field A as an advanced expression like the one below</p> <p style="text-align: center;">2 + 2</p> <p>and you set the result format to text, SmartList To Go would display the result 4 but it would treat it as a literal character rather than a numerical value. If you created Field B as an advanced expression like the one below</p> <p style="text-align: center;">Field A + 2</p> <p>the result would be 42, not 6, because SmartList To Go will treat the result of Field A as text based on the result format you selected.</p>

5.4.2.3 Notes on Advanced Expression Syntax

The Advanced Expression field type requires that you follow these syntax conventions when designing your expressions.

The Use of Brackets Around Field Names

In SmartList To Go Desktop only, the name of a field in an Advanced Expression may be enclosed by square brackets ([and]), as in the example:

Sqrt([length A] * [length A] + [length B] * [length B])

Square brackets are required in SmartList To Go Desktop when the name of the field includes a space. For example, if you were to design an expression that contained the field name length A and you failed to enclose the field name in square brackets, SmartList To Go may mistakenly interpret the name to be two separate fields name length and A.

Square brackets are optional in SmartList To Go Desktop when the name of the field does not include a space, as in the field name customerAge.

SmartList To Go Handheld does not use square brackets in this manner ever. If you design an advanced expression on the desktop that uses brackets and later view that expression on your handheld device, the square brackets will not appear. For example, the sample expression described above would simultaneously appear on the desktop as:

$\text{Sqrt}([\text{length A}] * [\text{length A}] + [\text{length B}] * [\text{length B}])$

and on the handheld as:

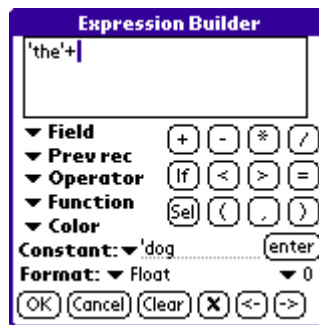
$\text{Sqrt}(\text{length A} * \text{length A} + \text{length B} * \text{length B})$

The Use of Single Quotes Around String (text) Values

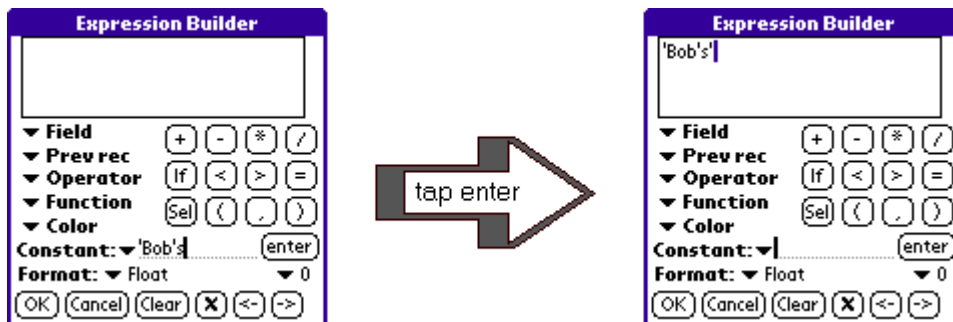
When your advanced expression contains a text string (text), you must enclose the text in single quotation marks ('), as in the example:

$\text{'This text'} + \text{'is enclosed'}$

To enter a string (text) in SmartList To Go, you will use the Constant input in the Expression Builder screen. You will enter the first single quote, enter the string (text) and tap the **enter** button. The SmartList To Go handheld application automatically enters the final single quote.



To enter a string (text) in SmartList To Go Handheld that includes single quotes, simply enter the first quote, then enter the string (text) exactly as you would like it to appear, as in the example below.



To enter a string (text) in SmartList To Go Desktop, type the string (text) exactly as you would like it to appear with the addition of single quotes at the beginning and end. Unlike SmartList To Go Handheld, SmartList To Go Desktop will not enter the end quote automatically.

To enter a string (text) in SmartList To Go Desktop that includes single quotes, it is necessary to precede the single quote with a backslash (\). To enter the string (text) **Bob's**, for example, you would type:

`'Bob\'s'`

5.5 Beaming

Many handhelds that use the PalmOS have built-in infrared emitter/receivers that enable users to "beam" data from one handheld to another without the need of a special cable or other direct connection. The SmartList To Go handheld application supports this capability.


In This Section . . .

- **5.5.1 SmartLists**
- **5.5.2 Records**

5.5.1 Beaming a SmartList

To beam one or more SmartLists to another handheld, you must have the SmartList To Go handheld application installed on the handheld you are beaming from. You do not need to have the SmartList To Go app installed on the handheld you are beaming to; you will have the option of installing the SmartList To Go handheld app as you beam the SmartLists.

To beam a SmartList between handhelds, position the handhelds so that their infrared emitter/receivers are facing each other, start from the SmartList To Go Home screen and follow these steps:

1. Click on the Beam icon  to open the Beam screen.
2. If the handheld to which you are beaming the SmartList already has the SmartList To Go handheld app installed, do not check the **Beam SmartList To Go application** checkbox. If the handheld to which you are beaming the SmartList does not already have the SmartList To Go handheld app installed, check this box. If the SmartList To Go handheld application is not installed, you will not be able to open, view or use the SmartList in any way.
3. Below the **Beam SmartList To Go application** checkbox you will find a list of all the SmartLists currently installed on your handheld. Check only those SmartLists you wish to beam to the new handheld.
4. Click the **Beam** button to initiate the beaming connection, or click **Cancel** to close the Beam window without sending any information.

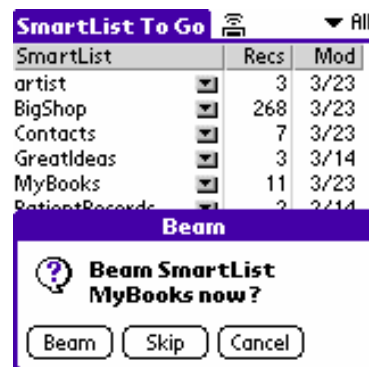
A small window will open describing the progress of the beaming procedure. First, it will indicate that your handheld is preparing the SmartList To Go handheld application, then that it is searching for the receiver in the receiving handheld.



The Beam screen



The Beam window



The Beam Cancellation screen

As with any beaming procedure in any handheld application, if the handhelds are unable to acquire a signal from each other, they will be unable to complete the transaction. This is most often due to the alignment and distance between the handhelds. For more information on the optimal alignment and distance between handhelds, please refer to that handheld's documentation.

5. When the transfer is complete, the receiving handheld will prompt you if you wish to accept the new files. Click **OK** to accept and install the new application and database data, or click **Cancel** to end the transaction without accepting the new data.

If the handhelds were unable to make a good connection, a small window will appear indicating the operation was canceled. This window will also appear if you manually cancel the beaming operation. Click **OK**.

A new window will open with options on how to proceed. Click **Beam** to try the operation again, click **Skip** to defer the operation until later, or click **Cancel** to abort the operation entirely.

The Beaming operation is now complete. The SmartList is now available on the receiving handheld.

5.5.2 Beaming a Record

In addition to beaming entire SmartLists, the SmartList To Go handheld application can also beam individual records from a SmartList between handhelds.

Unlike when beaming a SmartList, both the sending and receiving handhelds must already have the SmartList To Go handheld app installed when you beam a record. If the receiving handheld does not, remember that you can install it quickly and easily while beaming a SmartList using the directions in the section above.

Further, the receiving handheld must contain a SmartList with an identical field structure in order to accept the incoming record with its data intact. Unless you are already sure that the sending and receiving SmartLists already have an identical structure (for example, they may have both been installed from the same SmartList or Access database on the desktop), by far the best way to ensure that the receiving SmartList's structure is appropriate, you should follow this process outline:

1. Clone the SmartList on the sending handheld with all its records.
2. Delete extraneous records, if any, leaving only those records you wish to send to the receiving handheld.
3. Beam the clone SmartList you created in the last step to the receiving handheld. Choosing to beam a SmartList instead of records means that you are creating a brand new SmartList on the receiving handheld, guaranteeing that its structure is in complete agreement with the SmartList on the sending handheld.
4. Now that you have created a corresponding SmartList on the receiving handheld, you can delete the clone SmartList on the sending handheld - leaving the original with all its records intact - and in the future you can beam records back and forth directly.

To beam an individual record, begin in the Record screen of the individual record you want to beam, position the handhelds so that their infrared emitter/receivers are facing each other, and

follow these steps:

1. Open the Record screen **Menu** and select **Actions > Beam Record**.

A small window will open describing the progress of the beaming procedure. First, it will indicate that your handheld is preparing the SmartList To Go handheld record, then that it is searching for the receiver in the receiving handheld. As with any beaming procedure in any handheld application, if the handhelds are unable to acquire a signal from each other, they will be unable to complete the transaction. This is most often due to the alignment and distance between the handhelds. For more information on the optimal alignment and distance between handhelds, please refer to that handheld's documentation.

2. If the receiving handheld contains a SmartList with the same name as that within the sending handheld, the SmartList To Go handheld application will automatically put the record in that SmartList. If the receiving handheld does not have a SmartList of the same name, a small window will open prompting you to select the desired SmartList from a pull-down list. Open the pull-down list and select the SmartList you wish to add the record to.

If the handhelds were unable to make a good connection, a small window will appear indicating the operation was canceled. This window will also appear if you manually cancel the beaming operation. Click **OK**.

A new window will open with options on how to proceed. Click **Beam** to try the operation again, click **Skip** to defer the operation until later, or click **Cancel** to abort the operation entirely.

The Beaming operation is now complete. The record is now available in the receiving SmartList.

For information on beaming multiple records, see Section 5.8 Filtered Operations.

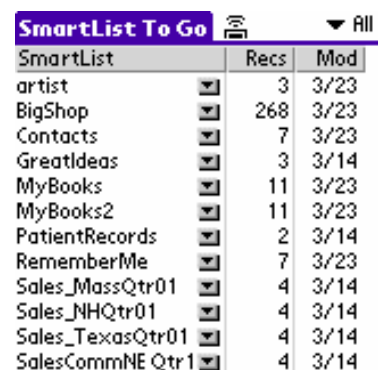
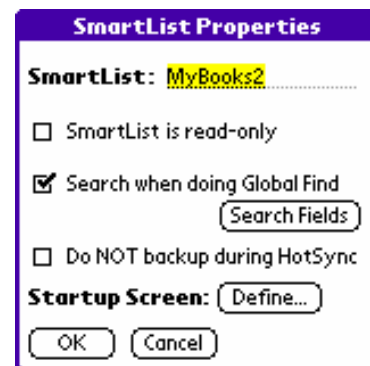
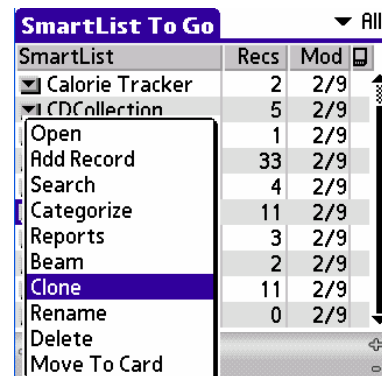
5.6 Cloning a SmartList

Cloning a SmartList makes a copy of the entire database structure, including fields and their properties, categories, lists and forms in a single process. During this process you have the option of creating the new clone with or without records.

To clone a SmartList, begin in the SmartList To Go handheld app Home screen and follow these steps:

1. Open the pull-down menu corresponding to the SmartList you wish to clone and select **Clone**. SmartList To Go will create a copy of the SmartList and automatically open the SmartList Properties screen.
2. Set the properties for the new SmartList as described in Section 4.1.2 Setting SmartList Properties. The new SmartList cannot have the same name as the original, so be sure to highlight the original SmartList name and enter a new name.
3. Click **OK** to implement these changes. Do not click **Cancel** to close the SmartList Properties screen without implementing your changes, because you must at least change the SmartList name.
4. A Confirmation window will open to indicate that the structure of the new SmartList has been created successfully and to ask if you wish to copy the records as well. Select **Yes** to copy of all the records in the original SmartList in the new clone SmartList. Select **No** to create the new SmartList without any records at all.

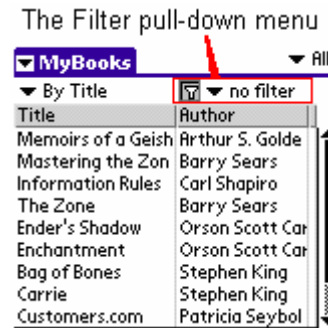
You will now return to the SmartList To Go handheld app Home screen, where you will find the new SmartList is added to the list of SmartLists.



5.7 Filters

A filter is a tool for creating a limited list of records based upon the information entered in one or more fields within each record. When you create a filter, you indicate that you want the SmartList To Go handheld application to compare the value in a certain field in every record and create a list of those records that have the value you are looking for.

For example, look at the SmartList MyBooks. Open the Filter pull-down menu in the top right corner of the screen, just below the Category menu, and notice that among the filters available are two called Borrowed and Top-Rated.



If you select the **Borrowed** filter, the SmartList To Go handheld application will check the Status field in every record and create a list that includes only those records in which the Status indicated is "Borrowed."

If you select the **Top-Rated** filter, the SmartList To Go handheld application will check the Rating field in every record and create a list that includes only those records in which the Rating indicated is "*****."

In addition, you can also create lists that apply two or more filters simultaneously. For example, if you select the **Multiple Filters** in the Filter menu, it will open a screen in which you can select both Borrowed and Top-Rated. The SmartList To Go handheld application will check every record in the SmartList for those with both "Borrowed" indicated in the Status field and "*****" indicated in the Rating field.

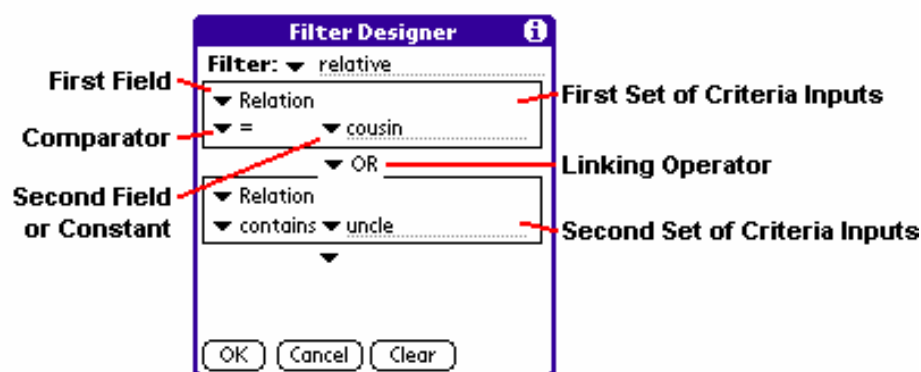
In This Section . . .

- 5.7.1 Configuring Filters
- 5.7.2 Applying Filters

5.7.1 Configuring Filters

The Filters Editor screen is your interface for working with filters. There are three ways to access the Filter Editor screen.

1. Open the Filters pull-down menu near the upper right corner of the screen and select the last option, **Edit Filters**.
2. Click on the **Tools** button at the bottom of the screen, the button with the hammer icon, to open the Tools pop-up menu and select Filter Editor.
3. Open the Tools pull-down menu from the Main Menu on the Home screen and select **Filters Editor**.



A filter works by comparing the value in a selected field to another value. This value may be the value of another field or it may be a constant value that you enter when you create the filter. You will use the Filters Editor screen to create this comparison. Those records that meet the criteria established by the filter will appear in the resulting view. Those records that do not meet the criteria will not. The SmartList To Go handheld application does not delete or alter the records it removes from view when applying a filter.

Consider, for example, that you have created a SmartList to track your collection of books. You have created fields that contain the book title, author's name, genre and edition. The edition field is a List type field that contains choices like "hardcover with dust jacket," "hardcover w/o dust jacket," "paperback," "audio on tape," "audio on CD" and "ebook."

Now consider that you want to create a filter that will show you only those books that were written by Stephen King. You would open the Filters Editor screen and construct the following filter:

Author = "Stephen King"

Such a filter, when applied, would test the values entered in the Author field against the constant value "Stephen King," and the SmartList To Go handheld app would only display those records in which the value was exactly that constant.



However, you are not limited to using only the equals sign as your comparator. When you create a filter, you can choose a variety of comparator's to create your filter criteria. The table below lists and describes these comparators.

=	The equals comparator checks that the first field matches the second field or constant exactly.
<>	The does not equal comparator checks that the first field does not match the second field or constant exactly.
<	The less than comparator checks that the first field is less than the second field or constant. This comparator creates a valid comparison for numerical values only; text values cannot be compared with the less than sign.

>	The greater than comparator checks that the first field is greater than the second field or constant. This comparator creates a valid comparison for numerical values only; text values cannot be compared with the greater than sign.
<=	The less than or equal to comparator checks that the first field is less than or equal to the second field or constant. This comparator creates a valid comparison for numerical values only; text values cannot be compared with the less than or equal to sign.
>=	The greater than or equal to comparator checks that the first field is greater than or equal to the second field or constant. This comparator creates a valid comparison for numerical values only; text values cannot be compared with the greater than or equal to sign.
contains	<p>The contains comparator checks that the first field contains the text entered in the second field or constant. The contains comparator is not case sensitive.</p> <p>In the example described above, the following filter</p> <p style="text-align: center;">Author contains "Ben"</p> <p>would return records in which the author's name contained the string (text) segment "Ben"- ignoring capitalization - whether the name is "Ben Bova," "Annette Benning" or "Gil Hibben."</p> <p>This comparator only creates a valid comparison for text values; numerical values cannot be compared with contains.</p>
Contains	<p>Like the contains comparator, the Contains comparator checks that the first field contains the text entered in the second field or constant. This comparator, however, is case sensitive.</p> <p>In the example described above, the following filter</p> <p style="text-align: center;">Author Contains "Ben"</p> <p>would return records in which the author's name contained the string (text) segment "Ben" including the capitalized B. This filter would accept the names "Ben Bova" and "Annette Benning," but not the name "Gil Hibben."</p> <p>This comparator only creates a valid comparison for text values; numerical values cannot be compared with Contains.</p>

starts	<p>The starts comparator is a more specific variation of the contains comparator. This comparator checks that the first field begins with the text entered in the second field or constant. The starts comparator is not case sensitive.</p> <p>In the example described above, the following filter</p> <p style="text-align: center;">Author starts "Ben"</p> <p>would return records in which the author's name began with the string (text) segment "Ben" - ignoring capitalization. This filter would accept names like "Ben Bova," "Benning, Annette," "benjamin franklin" and even just "Ben." This filter would not accept names like "Annette Benning" and "Gil Hibben."</p> <p>This comparator only creates a valid comparison for text values; numerical values cannot be compares with starts.</p>
Starts	<p>The Starts comparator, like starts, checks that the first field begins with the text entered in the second field or constant. This comparator, however, is case sensitive.</p> <p>In the example described above, the following filter</p> <p style="text-align: center;">Author Starts "Ben"</p> <p>would return records in which the author's name began with the string (text) segment "Ben" including the capitalized B. This filter would accept names like "Ben Bova," "Benning, Annette," and "Ben." This filter would not accept names like "Annette Benning," "benjamin franklin" and "Gil Hibben."</p> <p>This comparator only creates a valid comparison for text values; numerical values cannot be compares with Starts.</p>

A single filter can also test according to as many as three separate criteria using the linking operators **AND** and **OR**. When you select one of these linking operators, the SmartList To Go handheld application enables a second set of controls for setting filter criteria. If you select **AND**, then a record must satisfy both criteria for the filter to accept it. If you select **OR**, then a record must satisfy one or both criteria for the filter to accept it. If you then select another linking operator, the SmartList To Go handheld app will enable a third set of controls for setting filter criteria.

Continuing the example of the SmartList described above, consider that you wanted to create a view of all your audio books, regardless of whether they were on tape or CD. There is more than one way to configure a filter to accomplish this, but the following is one example.

Edition = "audio on tape"

OR

Edition = "audio on CD"



With such a filter, the SmartList To Go handheld application would display any record that met one or both of these criteria. In this example, both criteria work with the same field, Edition, but this does not have to be the case.

To create and configure a new filter, start in the Filters Editor screen and follow these steps. The rectangle near the top of the screen encloses the controls for setting the first filter criteria.

1. Open the Filter pull-down menu at the top of the screen and select an unused filter slot.
2. Erase the default name and enter the desired name with your stylus on the line provided.
3. Open the First Field pull-down menu and select the field whose values you wish to test. This pull-down menu contains the names of all the currently existing fields in the SmartList. You cannot enter a constant in this input.
4. Open the Second Field pull-down menu and select the field whose values you wish to use to test the value in the First field. This pull-down menu contains the names of all the currently existing fields in the SmartList. To compare the value in the First Field to a constant, select **constant**. This will insert a blank line in the input. Enter the text or numerical constant on this line with the stylus.
5. Open the Comparator pull-down menu and select how you want to compare the two values. Refer to the table above for a description of the available comparators.
6. If you would like to create another filter criteria for this filter, open the Linking Operator pull-down menu and select **AND** or **OR**, then repeat steps 3, 4 and 5. If you do not wish to create another criteria, continue to the next step.
7. Click **OK** to create the new filter or click **Cancel** to abort your changes and close the Filters Editor screen without creating the filter. If you click **OK**, the new filter's name will now appear on the Filters pull-down menu

The filter is now created.

To change the configuration of an existing filter, start from the Filters Editor screen. Open the Filter pull-down menu and select the name of the filter you want to change. Then follow steps three through seven from the process above.

5.7.2 Applying Filters

Use the Filters pull-down menu located near the top right corner of the screen to apply filters.

To apply a filter, begin in the SmartList's main screen. Open the Filters pull-down menu and select the name of the desired filter. The SmartList To Go handheld application will then test each record according to criteria described by that filter and remove any records that did not satisfy the criteria.

To apply multiple filters, begin in the SmartList's main screen. Open the Filters pull-down menu and select **Multiple**. This will open the Multiple Filters screen, which will display a checklist of all the currently existing filters. Check off only those filters you wish to apply. Then click **OK** to apply to the filters, or click **Cancel** to close the Multiple Filters screen without applying any filters.

To delete a filter, begin in the SmartList's main screen and follow these steps:

1. Open the Filters pull-down menu and select **Edit Filters** to open the Filters Editor screen.
2. Open the Filters pull-down menu and select the name of the filter you wish to delete.
3. Open the First Field pull-down menu in the first set of criteria inputs and select **none**. This will clear all the other inputs on the Filters Editor screen.
4. Click **OK** to apply your changes, or click **Cancel** to close the Filters Editor window without removing the filter.

The filter is now removed.

To add a default filter to a view, begin in the SmartList's main screen and follow these steps:

1. Open the Views pull-down menu and select **Edit Views** to open the View Designer screen.
2. Open the View pull-down menu and select the name of the view you wish to add the filter to.
3. Open the Default Filter pull-down menu. This menu contains a list of the currently existing filters. Select the filter you wish to add to the view.
4. Click **OK** to apply this change or click **Cancel** to close the View Designer screen without adding the default filter.

The default filter is now a part of the view. When you select this view, you will automatically apply the filter you selected as well.

5.8 Filtered Operations

All of the procedures we have covered in this manual so far have been performed on either an entire SmartList at a time or on single, individual records. Filtered Operations enable you to perform batch operations on select groups of records. You design what records to include in the group and then perform procedures such as:

- Deleting
- Checking or unchecking Checkbox fields
- Beaming
- Moving from one category to another
- Applying plug-ins

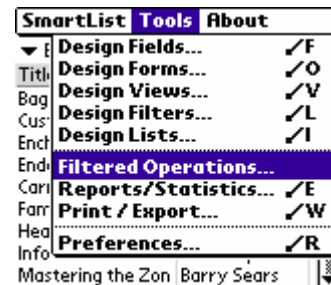
When you perform a Filtered Operation, you select the group on which to perform the operation by creating a List View that includes only those records you want to include. You can accomplish this with filters, as the name of this section suggests, but you can also use view designs and categories as well. The rule is the procedure will be performed on all records currently listed when you begin the filtered operation.

Once you have created this list, you will open the Filtered Operations screen, your primary interface for these procedures.

To perform a Filtered Operation, start from the List View screen of a SmartList and follow these steps:

1. Select a Filter, View, Category or some combination of these such that only those records you want to perform the operation on are listed. If the necessary Filters, Views and Categories do not yet exist, create them. For information on using these functions, see Sections 4.2.3 The View Editor Screen, 5.1 Categories and 5.7 Filters.
2. Open the Filtered Operations screen. To do this, either:
 - a. Open the List View Main Menu and select **Tools > Filtered Operations** -OR-
 - b. Click on the Tools icon to open a pull-down menu of tools and select **Filtered Operations**.

The Filtered Operations screen is your primary interface for performing these functions.





3. The buttons along the left side of the screen initiate each of the possible operations. The controls on the right side of the screen correspond only to the operation button on the same line, and they control how the operation works. This screen offers the follow operations:

- **Print/Exp** Select the **Print/Exp** button to print the selected records or export them to the PalmOS's built-in Memo Pad application and click **OK**. This will open the Print/Export screen. To print, you will require an infrared capable printer and appropriate printing software.
- **Delete** Select the **Delete** button to delete the selected records and click **OK**. A warning screen will open to verify that you wish to continue; deleting records cannot be undone. Click **OK** to continue or **Cancel** to abort the operation. This will return you to the List View screen.
- **Update** Select the **Update** button to check or uncheck Checkbox Type fields in the all the selected records. The two pull-down menus to the right of this button control this operation. If there is more than one Checkbox Type field in your records, select the name of the field in the pull-down menu on the left. To set the selected field to checked, select **ON** in the pull-down menu on the right; to set the selected field to unchecked, select **OFF** in the pull-down menu on the right. Then click **OK** to performing the operation and close the Filtered Operations screen or **Cancel** to close the screen without performing the operation.
- **Beam** Select the **Beam** button to beam the selected records from one handheld to another without the need of special cable or other physical connection. This function requires your device to have an infrared receiver/emitter; most handhelds that use the PalmOS come equipped with one. Click **OK** to initiate the Beaming procedure. For more information on how this procedure works, see Section 5.4 Beaming.
- **Move** Select the **Move** button to move the selected records from their current categories to the category you select in the pull-down menu to the right of the button. Select the destination category in that pull-down menu and click **OK** to move the records and return to the List View screen or click **Cancel** to close the screen without moving the records. For more information on categories, see Section 5.1 Categories.

- **Apply** Select the **Apply** button to apply the plug-in field you select in the pull-down menu to the right of this button and click **OK**. The exact function of applying a plug-in field depends upon the plug-in, so for more information refer to the plug-in's documentation.

The Filtered Operation is now complete.

5.9 Security

The SmartList To Go handheld application's security functions enable you use password protection to control users' abilities to make view or change SmartList data. Using the Security Settings screen, your primary interface for working with security, you can independently set passwords to control six different aspects of the SmartList To Go handheld app's functionality. Among these are:

- **Open SmartList** Applying a password to Open SmartList prevents a user who does not have the correct password from opening and viewing the data contained in a SmartList.
- **Record update** The Record update password does not prevent users from viewing a SmartList or its data. Rather, this password prevents a user who does not have the correct password from changing any of the data in a record.
- **Design change** Applying a password to Design change prevents a user who does not have the correct password from being able to add or remove fields or alter their properties, alter form designs or change list configurations.
- **View change** Applying the View change password prevents a user who does not have the correct password from being able change the design of views. They cannot alter add or remove views and they cannot alter existing views.
- **Beaming** Applying the Beaming password prevents a user who does not have the correct password from being able to beam the SmartList or any of its individual records to another handheld.
- **Copy** Applying the Copy password prevents a user who does not have the correct password from being able to clone the SmartList.

How Passwords Work

Passwords are always specific to a single SmartList. When a user attempts to perform a function that is protected by password, a small window will open prompting the user for the password.

If the user enters the correct password, the function is performed. In addition, SmartList To Go will not prompt the user again for subsequent uses of that function while the SmartList remains open, so you only have to enter the password once. Closing the SmartList, however, resets the passwords, so that to repeat a passworded function again after you have closed and reopened a SmartList will require you to enter the password again.

If the user enters the incorrect password, the function will be canceled.

In this Section . . .

- **5.9.1 Setting a Password**
- **5.9.2 Changing a Password**
- **5.9.3 Removing a Password**

5.9.1 Setting a Password

To set a password, begin in the List View screen and follow these steps:

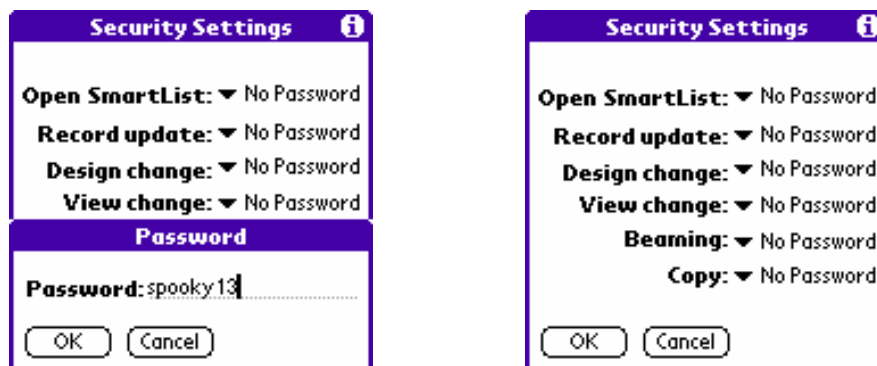
1. Open the **Main Menu** and select **SmartList > Security Settings** to open the Security Settings screen.



2. The six functions you can password protect are set to No Password by default. Open the Password pull-down menu that corresponds to the function you wish to protect and select **Assigned**. A small window will open in which to enter the desired password.
3. Enter the desired password in the Password window.



WARNING! Be careful when you assign a password that you remember the password you have chosen. If you forget the password, there is no way to bypass it and regain access to the passworded function.



Click **OK** to continue or **Cancel** to close the screen without applying the password.

4. A Verify Password window will open and ask you to enter the password a second time. Enter the password again and click **OK** or click **Cancel** to abort applying the password.

A Warning screen will then open to remind you to remember your password. Click **OK** to continue. This will return you to the Security Settings screen, which will now indicate **Assigned** for the function you assigned a password to.

5. Repeat steps 1 through 4 for each additional function you wish to control by password.

The password is now set.

5.9.2 Changing a Password

To change a password, begin in the List View screen and follow these steps:

1. Open the **Main Menu** and select **SmartList > Security Settings** to open the Security Settings screen.



The six functions you can password protect appear on the Security Settings screen. Functions that are currently restricted by a password will indicate **Assigned** in the corresponding pull-down menu to their right. Open the Password pull-down menu that corresponds to the function whose password you wish to change and select **Change**.

2. A small window will open in which to enter the current password. You must enter the current password before you can change it to a new password. Enter the Current Password and click **OK** or click **Cancel** to abort the password change.
3. A Password window will open in which to enter the new password. Enter the desired password in the Password window and click **OK** or click **Cancel** to abort the password change.



WARNING! Be careful when you assign a password that you remember the password you have chosen. If you forget the password, there is no way to bypass it and regain access to the passworded function.



5. A Verify Password window will open and ask you to enter the password a second time. Enter the password again and click **OK** to apply the password or click **Cancel** to abort applying the password.

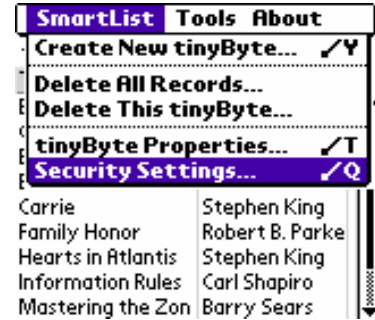
A Warning screen will then open to remind you to remember your password. Click **OK** to continue. This will return you to the Security Settings screen, which will now indicate **Assigned** for the function you assigned a password to.

6. Repeat steps 1 through 5 for each additional password you wish to change.

5.9.3 Removing a Password

To remove a password, begin in the List View screen and follow these steps:

1. Open the **Main Menu** and select **SmartList > Security Settings** to open the Security Settings screen.
2. The six functions you can password protect appear on the Security Settings screen. Functions that are currently restricted by a password will indicate **Assigned** in the corresponding pull-down menu to their right. Open the Password pull-down menu that corresponds to the function whose password you wish to remove and select **No Password**.
3. A small window will open in which to enter the current password. You must enter the current password before you can remove it. Enter the Current Password and click **OK** or click **Cancel** to abort the password removal.
4. An alert window will open to verify that the password is removed. Click **OK** to return to the Security Settings screen.
5. Repeat steps 1 through 4 for each additional password you wish to remove.



5.10 The Image Field

The Image Field enables you to store image data in your records exactly as you would text data or other kinds of data.

This field inserts an image icon onto the data entry form. To view or manipulate an image, you will open the record and click on the image icon. This will open the Image screen. You can use the Image screen to create, view or alter images.


The SmartList To Go handheld application supports three image formats: black and white (2 color), grayscale (16 colors) and 256 colors. 256 color mode is only available on handhelds with color screens. Images are 154x135 in dimension, and may range from 2K to 12K in size.

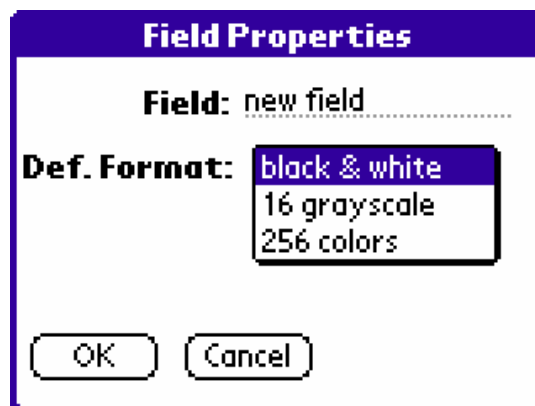
In This Section . . .

- 5.10.1 Creating and Configuring an Image Field
- 5.10.2 Using the Image Screen
- 5.10.3 Adding an Image Field to the Form

5.10.1 Creating and Configuring an Image Field

To create an image field, open the Field Editor and follow these steps:

1. Tap the **New** button, enter a **Name** for the image field and select Image from the **Type** dropdown menu.
2. Click on the **Field Properties icon**  to open the Image Field Properties screen.



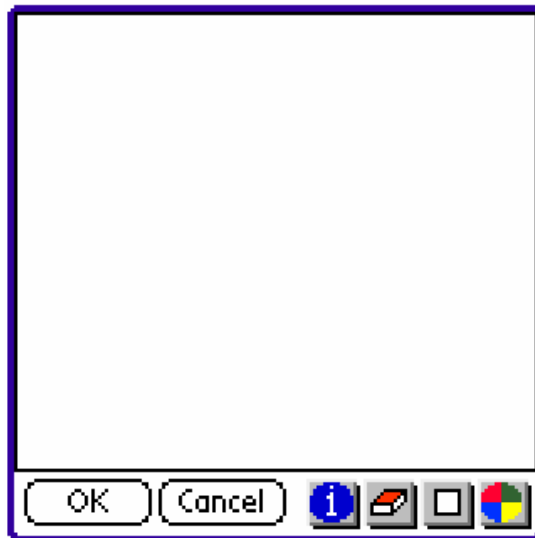
3. Select the image format in the **Def. Format** pull-down menu. If your handheld does not have a color screen, you may choose from black & white (2 color) or 16 shades of grayscale. If your handheld has a color screen, you may also choose 256 colors.

The Image field is now created. Remember to add the field to the data entry form according to the instruction provided in Section 4.2.2 The Form Designer Screen. The Image field will appear showing the Field name and an Image icon. To access an image, click on the Image icon to open the Image screen; the Image screen allows you to view, create and edit images.


5.10.2 Using the Image Screen

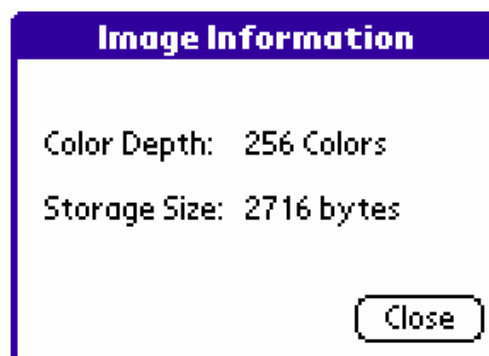
The Image screen is your primary interface for viewing, creating and editing images.


To open the image screen from the Record screen, click on the icon in the Image field.



The Image screen consists of two parts. The Image Viewing area dominates most of the area of the screen. The Image Toolbar, located at the bottom of the screen, provides the following controls for working with the image.


- **OK** - To save the currently displayed image in the Image field, click **OK** to close the Image screen and return to the Record screen.
- **Cancel** - To close the Image screen without applying the current changes, click **Cancel**. This will return you to the Record screen.
-  **Image Information** - Click the Image Information button to open Image Information Window. This window provides information on the image's size and color format, as in the sample below.

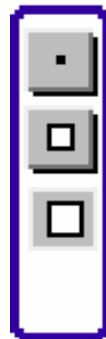



-  Image Tool - This button opens the Image toolbox. Select a tool from this box and the toolbox will close automatically. The Image Tool button will display the currently selected tool. Then use the tool in the Image Viewing area to create the desired image. Use the Image Toolbox as necessary to switch tools. From left to right, starting with the top row, the toolbox offers the following tools:



- Pencil - Use this tool to draw free-hand in the Image Viewing Area with the stylus.
 - Eraser - Use this tool to erase on the Image Viewing Area
 - Line Tool - Use this tool to draw straight lines. Place the stylus on the screen to indicate one end point of the line, then drag to the other endpoint of line and lift the stylus.
 - Circle Tool - Use this tool to draw circles. Place the stylus on the screen to indicate the center point of the circle, then drag to the desired radius and lift the stylus.
 - Filled Circle Tool - Use this tool to draw circles filled with solid color.
 - Text Tool - Use this tool to insert text into the image. Click with the stylus anywhere in the Image Viewing Area to indicate the text insertion point, the enter text as normal.
 - Rectangle Tool - Use this tool to draw rectangles/ Place the stylus on the screen to indicate one corner of the rectangle, then drag to the opposite corner and life the stylus to indicate the rectangle's height and width.
 - Filled Rectangle Tool - Use this tool to draw rectangles filled with solid color.
 - Clear Tool - Use this tool to clear the current image. An alert window will open to verify that you want to clear the image.

-  Image Tool Size - This button opens the Image Tools Size menu. Once you have selected a tool in the Image Toolbox, select the size of the tool using this toolbox.



-  Image Color Selection - If your handheld has a color screen, this button will open the Color Selection screen. This screen enables you to select the Pen and Text colors for your image.

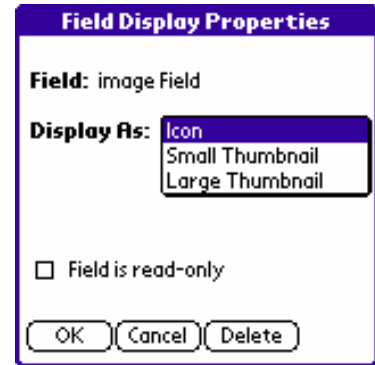


To use this screen, click on the color swatch that corresponds to the color you wish to change. This will open the Color Palette screen. Select the desired color and click **OK**. When you have selected your colors, click the **Done** button on the Color Selector screen.

5.10.3 Adding an Image Field to the Form

To add an image field to a form, open the Form Designer and follow these steps:

1. Click the line on the form where you would like to add the image field. This will open a pull-down list of existing fields.
2. Select the name of the image field. This will add the field to the form.
3. Click on the field label and open the Label Properties screen.
4. Enter the desired field label in the input labeled Field.
5. Select the desired font for displaying the label on the on the form.
6. Click **OK** to apply these changes to the label and return to the Form Designer Screen.
7. Click on the field type to open the Field Properties screen.
8. Select an image display style from the Display as pull-down menu. This menu offers three options for displaying images in a form:
 - **Icon** - This option inserts an icon in the Image Field. Users will click on this icon to open the Image Screen, where they can view the image full-size and modified it as necessary.
 - **Small Thumbnail** - This option inserts a thumbnail of the image that is 50 pixels by 45 pixels, or 33% of the image's full size. A small thumbnail takes up the equivalent of about four lines on the form. Users can click on the thumbnail to open the Image screen, where they can view the image full-size and modify it as necessary.
 - **Large Thumbnail** - This option inserts a thumbnail of the image that is 124 pixels by 108 pixels, or 80% of the image's full size. A large thumbnail takes up the equivalent of about 8 or 9 lines on the form. Users can click on the thumbnail to open the Image screen, where they can view the image full-size and modify it as necessary.
9. Click **OK** to apply your changes and close the Field Display Properties screen or click **Cancel** to close the screen without applying your changes.



The Image field is added to the form.

6. Relating SmartLists

Among the SmartList To Go handheld application's most powerful features are those that enable you to link a SmartList to other data sources like other SmartLists and the PalmOS's built-in Address Book application. These features are what make the SmartList To Go handheld application a relational database.

There are three main types of relational methods you will use to inter-relate your SmartLists. Each of them works in similar ways, allowing you to insert a link to information from another data into a record in the same way that you would insert a field. All three methods are, in fact, field types. They are:

- The DB Join and DB Lookup Field types
- The Address Join and Address Lookup Field Types
- The One-to-Many Field Type

The following sections describe each of these methods fully.

In This Chapter . . .

- **6.1 Relating SmartLists using DB Join and DB Lookup**
 - **6.1.1 Why use DB Join and DB Lookup Fields?**
 - **6.1.2 DB Join**
 - **6.1.3 DB Lookup**
- **6.2 Relating SmartLists to the Palm Address Book**
 - **6.2.1 Address Join**
 - **6.2.2 Address Lookup**
- **6.3 Relating SmartLists using One-to-Many**
 - **6.3.1 The One-to-Many Field**
 - **6.3.2 Enhancing a One-to-Many Relationship with DB Join and Key Fields**

6.1 Relating SmartLists using DB Join and DB Lookup Fields

In This Section . . .

- 6.1.1 Why use DB Join and DB Lookup Fields?
- 6.1.2 DB Join
 - 6.1.2.1 Creating a DB Join
 - 6.1.2.2 Using a DB Join
- 6.1.3 DB Lookup
 - 6.1.3.1 Creating a DB Lookup
 - 6.1.3.2 Using a DB Lookup

6.1.1 Why use DB Join and DB Lookup Fields?

The DB Join and DB Lookup Type fields work together to enable you to link a record in one SmartList to specific fields in another SmartList.

Consider the example of a SmartList that tracks Internet Service Providers and their corresponding contact information for a companies physical network. Each record would represent the group of servers at a single location. This would require you to create a SmartList that indicated information about the service provider such as the company's name, the company's address, the address of the location where the servers are hosted for you, the number of servers and so forth. You would also need information about the contact, the person you would call to address problems that arise, including the contact's name, title, email address, phone number, extension and more.

Take a look at the images below for how the SmartList might look.

Network Locations	Recd 3 of 4	Recd 3 of 4	Recd 3 of 4
View01 no filter	Unfi	Unfi	Unfi
Location ISP Contact	Location: Boston ISP: WinStar Contact: Andrew Lane	Location: Boston ISP: WinStar Contact: Andrew Lane	Location: Boston ISP: WinStar Contact: Andrew Lane
Dallas WinStar Dave Dunn	ISP Contact Location	ISP Contact Location	ISP Contact Location
Washington D. NetHome Bob Everet	Address: 17 Oak Street	Title: Site Manager	#: 4
Boston WinStar Andrew Lai	City, ST: San Diego, CA	Phone: 781-555-7268	Address: 95 Lynn St
New York WinStar Andrew Lai	Zip: 10234	Email: alane@winstar.com	City, ST: Boston, MA
	Phone: 781-555-1234		Zip: 1032
			Phone: 978-555-8572

If every ISP hosted only one group of servers and every contact only covered issues for a single group of servers, then you could probably maintain all this information in a single SmartList quite easily. However, it's more likely that you deal with the each company in a couple of different locations. You might deal with WinStar, for example, in locations in Los Angeles, San Francisco, Seattle, Boston, New York and Providence. That means that a number of records will have identical information in the fields that describe the company, and you will have to re-enter the information again for each record that applies to WinStar.

Similarly, a single contact might handle issues for all the WinStar locations in the Northeast: New York, Boston and Providence. That means the name and contact information for that contact will have to be entered 3 separate times.

You can see that this is not very efficient. To enter all your information, you have to repeat identical entries over and over, and if something changes - if WinStar, perhaps, changed their contact for the Northeast - you would have to make the change in a number of different locations.

A more efficient way to accomplish this would be to create separate SmartLists and link them together using DB Joins and DB Lookups. Your main SmartList describing server locations, referred to as the **Master SmartList**, would contain only the information that was specific to each location, but it would display information contained in the other SmartLists. One of the other SmartLists would describe only the ISP companies and their information, each company in a different record. The other would describe only the contacts and their information. The new SmartLists are referred to as **Detail SmartLists**. The new SmartLists might then include these fields:

Definition: A Master

SmartList is a SmartList that uses DB Join and DB Lookup fields as pointers to point to, or borrow, information from another SmartList.

Definition: A Detail SmartList

is a SmartList that stores data used by another SmartList. The Detail SmartList may also be useful on its own, or it may exist solely to provide data to a Master SmartList. The Master SmartList acquires data from the Detail SmartList using DB Join and DB Lookup fields.

Network Locations	ISP Companies
<ul style="list-style-type: none"> Location Location's Address Location's City and State Locations Zip Code Location's Phone Number Number of Servers at Location 	<ul style="list-style-type: none"> ISP Company's Name Company's Address Company's City and State Company's Zip Code Company's Phone Number
	ISP Contacts
	<ul style="list-style-type: none"> Contact's Name Contact's Title Contact's Phone Number Contact's Email address

In this configuration, the Network Locations SmartList will still display the same information, but it will not store repetitive information. Instead, it will contain pointers to this information in the ISP Companies and ISP Contacts SmartLists. Pointers refer to information in another SmartList and automatically retrieve and display it when you open the record in the Master SmartList.

This is far more efficient. If two locations have the same ISP Company, you will not have to enter the information twice; instead, you will aim the pointers in each record in the Master SmartList to the same record in the ISP Companies Detail SmartList. If contact information changes, you will not have to rewrite his information in every record in which he appears; instead you will change

his one entry in the ISP Contacts Detail SmartList, and every record that points to that contact will automatically reflect the changes.

6.1.2 DB Join

A DB Join is a field in a SmartList record that both establish a link to a specific field in a record in another SmartList and displays data from that field. Instead of storing data as other fields do, it creates a pointer to another field and whenever the record is opened, it automatically follows the pointer, retrieves the necessary data and displays it in the record.

For example, in the example above you would create two DB Joins, one each for the Detail SmartLists. The first would be a DB Join to the ISP Companies SmartList; the second would be a DB Join to the ISP Contacts SmartList.

6.1.2.1 Creating a DB Join

When you create SmartLists that will be connected by DB Joins, first create both the Master and Detail SmartLists, then populate at least the Detail SmartList with data before creating the link between them.

To create a DB Join, start from the Field Editor in the Master SmartList and follow these steps:

1. Tap the **New** button to open the **Single Field Editor**.
2. Enter a **Name** for the field and select **DB Join** from the **Type** dropdown list.
3. Click on the Field Properties icon to the right of the field type to open the Field Properties screen.
4. Open the **Source SL** pull-down menu. You will see a list of all the other SmartLists that currently exist on your handheld. Select the SmartList that will serve as the Detail SmartList.

The screenshot shows the 'Single Field Editor' dialog box. At the top, it says 'SmartList To Go' followed by a dropdown arrow and 'Field Editor'. The dialog has a title bar 'Single Field Editor'. Inside, 'Name:' is followed by 'Status'. 'Type:' is followed by a dropdown arrow and 'DB Join'. To the right of 'DB Join' is a small square icon with three dots. Below this is a text box with the text: 'The DB Join field enables you to inter-relate SmartLists. Rather than storing a value directly, this field contains a pointer to'. To the right of this text box is a vertical double-headed arrow. At the bottom are two buttons: 'Done' and 'Cancel'.


The screenshot shows the 'Field Properties' dialog box. It has a title bar 'Field Properties'. Inside, 'Field:' is followed by 'Status'. 'Source SL:' is followed by a dropdown arrow and 'My Books'. 'Join:' is followed by a dropdown arrow and 'Last Pg'. 'Display:' is followed by a dropdown arrow and 'Pages'. 'Use view:' is followed by a dropdown arrow and 'By Author/Titl'. At the bottom are two buttons: 'OK' and 'Cancel'.

5. Open the **Join** pull-down menu. You will see a list of all the fields in the Detail SmartList to which it is possible to join. Select the Field that contains the data you want to retrieve from the Detail SmartList.
6. Open the **Display** pull-down menu. As in the case of the Join pull-down menu, you will see a list of the fields in the Detail SmartList. Select the field whose data you wish to display in the DB Join field in the Master SmartList.



Note: The field you Join to is often the same field you Display, but they do not have to be. It is a common practice to create Detail SmartLists with Key fields, to Join to the Key field but to display another field whose data is more meaningful to the user. In the example above, you might Join to the Key field, but display the Company Name field. This is useful for reducing the possibility of confusion in complex inter-relationships between SmartLists.

7. Open the **Use View** pull-down menu. You will see a list of the Views that are currently configured in the Detail SmartList. Select the View you wish to see when you are selecting the record whose data the DB Join will point to.
8. Click **OK** to close the Field Properties screen and apply these properties or click **Cancel** to close the screen and abandon these properties. This will return you to the Field Editor.


The DB Join field is now created. Add the DB Join to the form just as you would any other field. The field will appear as a text field with a Link icon  next to it.

6.1.2.2 Using a DB Join

In most fields, you enter the data or select an item directly. In a DB Join field, you instead select a record from the Detail SmartList that will supply data to the DB Join field.

Before you can use a DB Join field, the Field Properties must first be configured as described in the section above.

To use a DB Join field, start from the Record screen and follows these steps:

1. Click on the **Link** icon  to the right of the DB Join field. This will open a temporary view of the Detail SmartList the DB Join is linked to. This view of the Detail SmartList looks exactly like what you would see if you had opened the SmartList normally, and it uses the View you selected in the Use View field when setting up the DB Join's properties.
2. Select the record in the Detail SmartList that contains the data you wish to use. The temporary view of the Detail SmartList will close and you will return to the Record screen in the Master SmartList. The DB Join field will retrieve and display data from the record you selected according to the Field properties you configured when you created the field. In addition, any DB Lookups that are associated with it, if any, will also retrieve and display the appropriate data.

The process is now complete.

6.1.3 DB Lookup

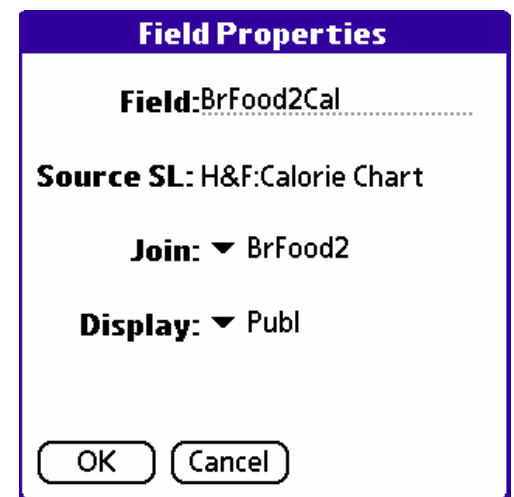
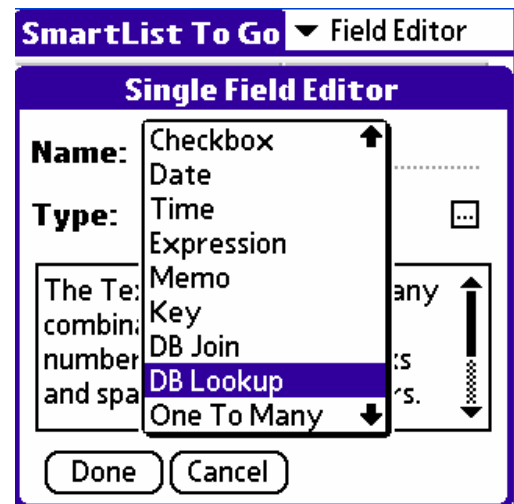
Creating a DB Join retrieves information from just one field in another SmartList. Once that connection between records is established, you can use DB Lookups to retrieve information from additional fields using the same connection.

DB Lookups always use the link established by a DB Join, so you must create a DB Join before you can create a DB Lookup.

6.1.3.1 Creating a DB Lookup

To create a DB Lookup, start from the Field Editor in the Master SmartList and follow these steps:

1. Tap the **New** button to open the **Single Field Editor**.
2. Enter a **Name** for the field and select **DB Lookup** from the **Type** dropdown list.
3. Click on the Field Properties icon to the right of the field type to open the Field Properties screen.
4. Open the **Source SL** pull-down menu. You will see a list of other SmartLists the Master SmartList is currently linked to by a DB Join. Select the SmartList that contains the data you will require.
5. Open the **Join** pull-down menu. You will see a list of all the DB Join fields in the Master SmartList that have established links the DB Lookup can use. Select the DB Join field that links to the Source SmartList you selected in the last step.
6. Open the **Display** pull-down menu. You will see a list of the fields in the Detail SmartList. Select the field whose data you wish to display in the DB Join field in the Master SmartList.
7. Click **OK** to close the Field Properties screen and apply these properties or click **Cancel** to close the screen and abandon these properties. This will return you to the Field Editor.



The DB Lookup field is now created. Add the DB Lookup to the form just as you would any other field. The field will appear as a text field without line for entering the text.

6.1.3.2 Using a DB Lookup

A DB Lookup field, properly created and configured, does not require any action on your part to acquire data. When you select the record in the Detail SmartList that will supply data to a DB Join, that selection applies to any DB Lookups that are associated with it. The DB Lookups will automatically retrieve and display the data you require from the record you selected in the DB Join.

6.2 Relating SmartLists to the Palm Address Book

The Address Join and Address Lookup fields work almost exactly like the DB Join and DB Lookup fields, except that they allow you to link a record in a SmartList to an entry in the PalmOS's built-in Address Book application. To continue the example from the last section, you might decide to use an Address Join and Address Lookup fields instead of creating a Detail SmartList to store information about the ISP Contacts.

Please read Section 6.1 Relating SmartLists using DB Join and DB Lookup before proceeding with this section.

In This Section . . .

- **6.2.1 Address Join**
 - **6.2.1.1 Creating an Address Join**
 - **6.2.1.2 Using an Address Join**
- **6.2.2 Address Lookup**
 - **6.2.2.1 Creating an Address Lookup**
 - **6.2.2.2 Using an Address Lookup**


6.2.1 Address Join

Similar to the DB Join, the Address Join established the link between the record in the Master SmartList and the entry in either the built-in Address Book or DataViz's Beyond Contacts contacts list. Once the link is established it retrieves and displays information from one of the fields in that entry.

6.2.1.1 Creating an Address Join

To create an Address Join, start from the Field Editor screen and follow these steps:

1. Tap the **New** button to open the **Single Field Editor**.
2. Enter a **Name** for the field and select **Address Join** from the **Type** dropdown list.
3. Click on the Field Properties icon to the right of the field type to open the Field Properties screen.
4. In the **Look In** field, choose either Beyond Contacts or Palm Address Book. (Note: to use Address Join with Beyond Contacts, you must have Beyond Contacts installed).
5. Open the **Display** pull-down menu. You will see a list of possible data combinations and formatting styles for the information in the Address Book entry. Select the format you wish to appear in the Address Join field in the Master SmartList.
6. Click **OK** to close the Field Properties screen and apply these properties or click **Cancel** to close the screen and abandon these properties. This will return you to the Field Editor.


The Address Join field is now created. Add the Address Join to the form just as you would any other field. The field will appear as a text field with a **Link** icon  on the right side.

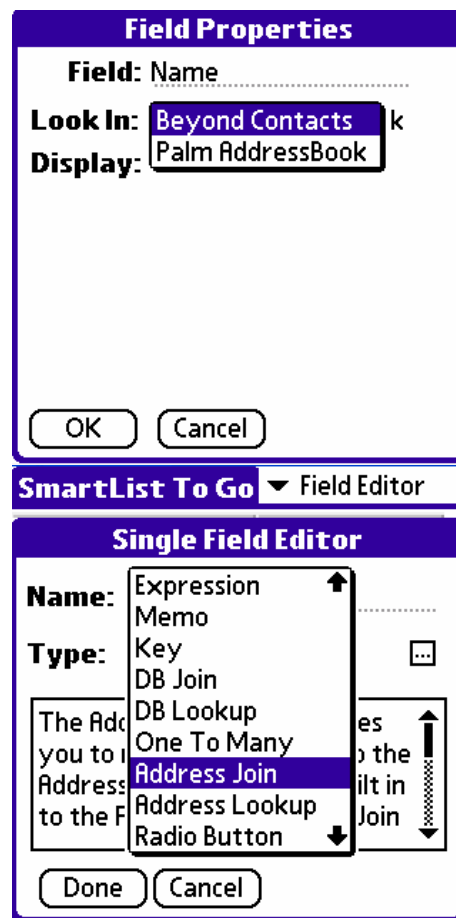
6.2.1.2 Using an Address Join

In most fields, you enter the data or select an item directly. In the Address Join, as in the DB Join field, you instead select the record that will supply the data for the field. In an Address Join, that record is an entry in the Address Book application.

Before you can use an Address Join field, the Field Properties must first be configured as described in the section above.

To use an Address Join field, start from the Record screen and follows these steps:

1. Click on the **Link** icon  to the right of the Address Join field. This will open a temporary view of your Address Book or Beyond Contacts contacts list.



The image shows two screenshots of the application's interface. The top screenshot is the 'Field Properties' dialog box. It has a title bar 'Field Properties' and a 'Field: Name' label. Below it, 'Look In:' is set to 'Beyond Contacts' and 'Display:' is set to 'Palm AddressBook'. There are 'OK' and 'Cancel' buttons at the bottom. The bottom screenshot is the 'Single Field Editor' dialog box. It has a title bar 'Single Field Editor'. The 'Name:' field is empty. The 'Type:' dropdown menu is open, showing a list of options: Expression, Memo, Key, DB Join, DB Lookup, One To Many, Address Join (which is highlighted), Address Lookup, and Radio Button. There are 'Done' and 'Cancel' buttons at the bottom.

2. Select the contacts entry that contains the data you wish to use. The temporary view of the Address Book or Beyond Contacts will close and you will return to the Record screen in the Master SmartList. The Address Join will retrieve and display data from the contacts entry you selected according to the properties you configured when creating the field. In addition, any Address Lookups that are associated with it, if any, will also retrieve and display the appropriate data.

The process is now complete.

6.2.2 Address Lookup

An Address Join retrieves a limited amount of information from an Address Book or Beyond Contacts entry, enough to fill just one field on the SmartList Record. The Address Lookup uses the link established by the Address Join to retrieve additional data.

Address Lookups always use the link established by an Address Join, so you must create an Address Join before you can create an Address Lookup.

6.2.2.1 Creating an Address Lookup

To create an Address Lookup, start from the Field Editor in the Master SmartList and follow these steps:

1. Tap the **New** button to open the **Single Field Editor**.
2. Enter a **Name** for the field and select **Address Lookup** from the **Type** dropdown list.
3. Click on the Field Properties icon to the right of the field type to open the Field Properties screen.
4. Open the **Join** pull-down menu. You will see a list of the Address Join fields in the Master SmartList that have established links the Address Lookup can use. Select an Address Join to link the Address Lookup to it.
5. Open the **Display** pull-down menu. You will see a list of possible data combinations and formatting styles for the information in the Address Book entry. Select the format you wish to appear in the Address Lookup field in the Master SmartList.
6. Click **OK** to close the Field Properties screen and apply these properties or click **Cancel** to close the screen and abandon these properties. This will return you to the Field Editor.

The Address Lookup field is now created. Add the Address Lookup to the form just as you would any other field. The field will appear as a text field without line for entering the text.

6.2.2.2 Using an Address Lookup

An Address Lookup field, properly created and configured, does not require any action on your part to acquire data. When you select the record in the Detail SmartList that will supply data to an Address Join, that selection applies to any Address Lookups that are associated with it. The Address Lookups will automatically retrieve and display the data you require from the record you selected in the Address Join.

6.3 Relating SmartLists using One-to-Many

In Section 6.1 Relating SmartLists using DB Join and DB Lookup, you learned that DB Join fields were a method of linking one record in a Master SmartList to one record in a Detail SmartList. You can link any number of Master SmartList records to a single record in the Detail SmartList, and you can link a Master SmartList record to more than one record in more than one Detail SmartList by creating multiple DB Join fields. However, each DB Join represents a link that is always one-to-one.

Using a One-to-Many field, you can link a single Master SmartList record to many records in a Detail SmartList.

In This Section . . .

- **6.3.1 The One-to-Many Field**
 - **6.3.1.1 Creating a One-to-Many Relationship**
 - **6.3.1.2 Using a One-to-Many Field**
- **6.3.2 Enhancing a One-to-Many Relationship with DB Join and Key Fields**

6.3.1 The One-to-Many Field

When you create a One-to-Many field, you select which SmartList it will use as a Detail SmartList. The One-to-Many field appears as a button on your form. When you click on this button, SmartList To Go opens a temporary view of the Detail SmartList that includes only those Detail records that correspond to the Master record.

This association between records is based on matching criteria you configure when you create the field. You select a field in the Master SmartList and a field in the Detail SmartList; when you click the One-to-Many button in the Master record, it opens the Detail SmartList, and tests every record. If the data in the Detail record field matches the data in the Master SmartList field, the SmartList To Go handheld application displays the record; if not, it doesn't. This creates a List View in the temporary view that includes only those Detail records that match the criteria.

For example, consider that you were creating a SmartList called **FavAuthors** in which you will store information about your favorite authors. In this SmartList, you will create a One-to-Many field that creates a link to the SmartList **MyBooks** (the same SmartList we have used as an example throughout this manual). This link creates a button in each record that, when pressed, will open a temporary view of MyBooks that lists all books in MyBooks written by that author. In this case, the One-to-Many field must be configured to compare the Name field in FavAuthors to the Author field in MyBooks. Any MyBooks record in which Author matches Name in FavAuthors is displayed; those records that do not match are filtered out.

In the temporary view of MyBooks, you can open and view any record you wish. When you are finished, click on the back arrow just to the left of the Filters pull-down menu to return the Record screen in the Master SmartList.

You can further develop the matching criteria by selecting as many as three field matching relationships between the records in each SmartList.

6.3.1.1 Creating a One-to-Many Relationship

To create a **One-to-Many relationship**, start from the Field Editor and follow these steps:

1. Tap the **New** button to open the **Single Field Editor**.
2. Enter a **Name** for the field and select **One To Many** from the **Type** dropdown list.
3. Click the Field Properties icon next to One To Many to open the Field Properties Screen.
4. Open the **Related SL** pull-down menu at the top of the screen. You will see a list of the SmartLists currently on your handheld. Select the Detail SmartList that contains the data you require. In this example, described above, you would select MyBooks.

In the **Relationship** portion of Field Properties screen, you will create the criteria by which records are selected in the Detail SmartList.

4. Open the first pull-down menu under **This SL**. You will see a list of Master SmartList fields; select the field you wish to use as a basis for selecting Detail SmartList records. In the example described above, you would select the Name field.
5. Open the first pull-down menu under **Related SL**. You will see a list of Detail SmartList fields; select the field you wish to match against field you selected in the first **This SL** menu. In the example described above, you would select the Author field.
6. If desired, repeat Steps 4 and 5 for the second and third pull-down menus under **This SL** and **Related SL**. This will create additional criteria to compare the Master SmartList record to Detail SmartList records. A Detail record must meet all established criteria to be displayed.

For example, you might create a One-to-Many so that the first criteria compares the Name field in the Master to the Author field in the Detail and the second criteria compares the Publisher field in the Master to a Publisher field in the Detail. If you were in a Record for the author Robert Jordan, the Publisher field might contain the publisher name TOR. When you click the One-to-Many button, the SmartList To Go handheld application will compare the fields in the Robert Jordan record to the records in MyBooks. Perhaps a dozen records in MyBooks may match Name to Author, indicating that a dozen records in MyBooks describe books written by Robert Jordan.

Perhaps 20 or more records in MyBooks may match Publisher to Publisher, indicating the 20 records in MyBooks describe books published by TOR. However, if only six records match both Name to Author **and** Publisher to Publisher, only these six records will appear in the List View in the temporary view of MyBooks. These records describe only those books that were both written by Robert Jordan **and** published by TOR.



Note: In order for a match to occur, the data in each field in the comparison must match exactly. This includes exact spelling and capitalization. If there is any difference in the data at all, such as an accidental misspelling, the Detail record will not appear on the list. To avoid possible difficulties, we highly recommend the use of a DB Join field and even the use of Key fields to ensure data agreement. For more information on how to enhance a One-to-Many relationship with DB Join and Key fields, see Section 6.3.2 Enhancing a One-to-Many Relationship with DB Join and Key Fields below.

7. Open the **View** pull-down menu on the Field Properties screen. You will see a list of the Views currently configured in the Detail SmartList you selected in **Related SL** in Step 3. Select the View you wish to see when you open the temporary List View of the Detail SmartList.
8. Create the title of the temporary view of the Detail SmartList using the **Label** and **Field** controls. Enter the text of the title in the text input Label, then Open the Field pull-down menu and select the Mater SmartList field you wish to use in the title. In the example described above, you might enter "Books written by " in Label and select the Name field in Field. If you clicked on the One-to-Many button in the Robert Jordan record in FavAuthors, you would see a temporary view of MyBooks titled "Books written by Robert Jordan."
9. Click **OK** to apply the configuration you have create, or click **Cancel** to close the Field Properties screen without applying the configuration. This will return you to the Field Editor.
10. Enter the desired name of the field in the Field column corresponding to the One-to-Many field.



The One-to-Many Field is now created. Place the field on the form as you would place any field (see Section 4.2.2 The Form Designer Screen). The field will appear as a button with the field's name on it.



6.3.1.2 Using a One-to-Many Field

To use a One-to-Many field, start from the Record screen and follow these steps:

1. Ensure that the fields in the record that are part of the comparison criteria used by the One-to-Many field contain their appropriate data.
2. Click on the One-to-Many field button. SmartList To Go will compare the data in selected fields in this Master record to the data in the selected fields in every Detail record. It will open a temporary view of the Detail SmartList that looks almost exactly the same as though you had opened the Detail SmartList normally with three exceptions. First, only those records that match the One-to-Many criteria will appear in the View List; SmartList To Go will have filtered out the rest. Second, the title you configured in the One-to-Many's Filed Properties screen will appear at the top of the screen just below the main tab. Third, a back arrow will appear at the top left corner of the screen, just left of the Views pull-down menu and just below the main tab. This arrow allows you to return to the Record screen in the Master SmartList.
3. Select and view and records in the Detail SmartList view exactly as you normally would. You may make changes to records or create new records as you wish.
4. When you are finished with the temporary view, click the **back arrow** button to return to the Record screen in the Master SmartList.

The process is now complete.

6.3.2 Enhancing a One-to-Many Relationship with DB Join and Key Fields

Perhaps the most difficult part of creating a One-to-Many relationship between SmartLists is ensuring that the data in the matching criteria fields that is supposed to match, matches exactly. The best way to ensure such matching is to use a DB Join field in the criteria. You create the DB Join field in the Detail SmartList and connect it to the matching field in the Master SmartList.

A way to further ensure proper matching is to create a Key field in the Master and join the Detail DB Join to the Key field. This DB Join would join to the Key field, but display the Name field, and the One-to-Many matching criteria would compare the Master Key field to the Detail DB Join field.

The table below illustrates how this configuration could be applied to our example of the FavAuthors and MyBooks SmartList. As you look at this table, remember that this is just an example; the specifics for your SmartLists will be different.

Master - FavAuthors	Detail - MyBooks
Key Field: AuthorKey	Text Field: Title
Text Field: Name	DB Join Field: Author <ul style="list-style-type: none">• Joined to AuthorKey• Displays Name
One-to-Many Field: Book by Author <ul style="list-style-type: none">• Compares AuthorKey to Author	

Appendix A. SmartList To Go Field Types

In This Appendix . . .

- **A.1 Supported Field Types**
- **A.2 Field Type Capabilities**

A.1 Supported Field Types

The SmartList To Go handheld application supports 21 field types, listed in the table below.

Field Name	Description	Fields Properties
Text	Stores string (text)s of up to 254 characters, numbers, spaces, punctuation and special characters.	- Field Name - Default value - Auto Writer - Do not capitalize first letter: disables automatic capitalization of the first letter in the field
Integer	Stores whole number values from -32,768 to 32,767.	- Field Name - Default Value - Increment/Decrement: inserts up and down arrows for changing values.
Long Integer, "Long"	Like Integer, stores whole number values. However, uses additional memory to stores values ranging from -2,147,483,648 to 2,147,483,647.	- Field Name - Default Value - Increment/Decrement: inserts up and down arrows for changing values.
Floating Point, "Float"	Stores numerical values with up to 3 digits after decimal point.	- Field Name - Default Value - Dec. Digits: format the values by selecting 0 to 3 digits after the decimal.
List	Enables the user to create a list of preset values and stores the selected value.	- Field Name (Configure the list of values in the List Editor screen)
Date	Stores date values.	- Field Name - Default Value - Record Added: Automatically stores the current date on the day the record was created. - Record Updated: Automatically stores the date the record was last updated.
Time	Stores time values.	- Field Name - Default Value - Record Added: Automatically stores the current time when the record was created. - Record Updated: Automatically stores the time the record was last updated.
Checkbox	Stores "True or False" values that display as a checkbox. Checked is True; unchecked is False.	- Field Name

Expression	Stores the result of a mathematical expression that can use constants, other fields in this records or other fields in the previous record as factors. Supports up to 8 operations and 3 levels of nesting (using other Expression fields as factors). Other fields used as factors must be created before designing the Expression.	<ul style="list-style-type: none"> - Field Name - Select the format of the result from a list of numerical, time and date formats (Configure Expression using the Expression Builder screen)
Advanced Expression	The Advanced Expression field stores a value determined by a mathematical or boolean expression that uses constants and one or more values from other fields as operands. In addition to calculating numerical values, this field can also concatenate text value, including converting a mathematical value to text automatically in order to concatenate with another text value.	<ul style="list-style-type: none"> - Field Name - Select the format of the result from a list of numerical, time and date formats (Configure Expression using the Expression Builder screen)
Memo	Stores longer string (text)s of text containing up to 4096 characters, numbers, spaces, punctuation and special characters.	<ul style="list-style-type: none"> - Field Name - Default value - Auto Writer - Do not capitalize first letter: disables automatic capitalization of the first letter in the field
Key	Stores numerical values as unique identifiers.	<ul style="list-style-type: none"> - Field Name
DB Join	Joins the current (Detail) SmartList to records in another (Master) SmartList and extracts and displays information from the Master SmartList record.	<ul style="list-style-type: none"> - Field Name - Source SL: the Master SmartList with which to Join - Join: the Master field whose data you want to extract into the DB Join field - Display: the Master field whose data you want to appear in the DB Join field (may be different from the data stored in the field) - Use View: the view in the Master SmartList to use when selecting the record with which to Join the Detail SmartList record
DB Lookup	Uses the link created by a DB Join to extract additional information from the Master SmartList.	<ul style="list-style-type: none"> - Field Name - Source SL: the joined SmartList from which to extract information - Join: the Master field whose data you want to extract into the DB Lookup field - Display: the Master field whose data you want to appear in the DB Lookup field (may be different from the data stored in the field)
One-to-Many	Links a record in the current (Detail) SmartList to a list of multiple records in another (Master) SmartList by selecting all records that have identical values in a certain field.	<ul style="list-style-type: none"> - Field Name - Related SL: the Master SmartList - Relationship: establishes the criteria for selecting records in the Master SmartList based on data in the Detail SmartList record. - View information: defines the appearance of the temporary list of Master SmartList records.
Address Join	Links a SmartList record to an entry in the PalmOS Address Book or to DataViz's Beyond Contacts.	<ul style="list-style-type: none"> - Field Name - Display: format data by choosing from 27 different displays

Address Lookup	Uses the link established by an Address Join to extract and display additional information from an entry in the PalmOS Address Book or Beyond Contacts (from DataViz).	<ul style="list-style-type: none"> - Field Name - Join: select the Address Book field with which to link. - Display: format data by choosing from 27 different displays
Radio Button	Enables the user to create a list of preset values and stores the single selected value. Unlike List, all values are always visible.	<ul style="list-style-type: none"> - Field Name - Buttons: selects the number of preset choices and records their values - Selects one of the buttons to be the default. If none is selected, the first button is the default.
Image	Stores an image instead of text or numerical values.	<ul style="list-style-type: none"> - Field Name - Define Format: select from black and white, various gray-scale formats or 256 colors.
Conditional Expression, "Conditional Expr"	Selects between to preset values based on the result of logical expression that is either True or False. This expression often relies on the values in other fields. If the expression is true, one preset is used; if false, the other is used. The preset values may also incorporate values from other fields.	<ul style="list-style-type: none"> - Field Name - Condition: design the logical expression that determines which preset value is used. - Then: design the value to use when the Condition is true. - Else: design the value to use when the Condition is false.
Auto Incremental	Stores an integer values that automatically increments by one from the value in the previous record added. Unlike Key, these values are changeable.	<ul style="list-style-type: none"> - Field Name
SmartList To Go Plug-In	Enables the user to expand SmartList To Go functionality by inserted Plug-Ins.	<ul style="list-style-type: none"> - Field Name - Plug-In: Select the installed Plug-In to insert.

A.2 Field Type Capabilities

For logical reasons, not all tools and functions can be applied to all field types. For example, you apply the Sort function to arrange records according to an Image field. Other examples, however, may not be as obvious, so the table below lists which tools and functions may be used in conjunction with each field type.

Field Type	View Display	Use With Filters	Sort	Find	Reports/Statistics	Use as Join Field in DB Join	Use as Display Field in DB Join or Lookup	Use in Expression	Use as Previous Value in Expression	Use in Adv Expression	Use asPrevious Value in Adv Expression
Text	Yes	Yes	Yes	Yes	No	Yes	Yes	No	No	Yes	Yes
Integer	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Long	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Floating Point	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes
List	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	No	Yes	Yes
CheckBox	Yes	Yes	Yes	Yes	Yes	No	Yes ¹	Yes	Yes	Yes	Yes
Date	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Time	Yes	No	Yes	No	Yes	No	Yes	Yes	Yes	Yes	Yes
Expression	Yes	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes	Yes
Memo	Yes	Yes	Yes	Yes	No	No	Yes	No	No	Yes	Yes
Key	Yes	Yes	Yes	Yes	No	Yes	Yes	No	No	No	No
DB Join	Yes	Yes	Yes	Yes	No	No	No	No	No	Yes	No
DB Lookup	Yes	No	Yes	Yes ²	No	No	No	Yes	No	Yes	No
One to Many	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Address Join	Yes	Yes	Yes	Yes	No	No	No	No	No	Yes	Yes
Address Lookup	Yes	Yes	Yes	Yes	No	No	No	No	No	Yes	Yes
Radio Button	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	No	Yes ³	Yes ³
Image	Yes	N/A	N/A	N/A	No	No	No	No	No	No	No
Condition Expression	Yes	Yes	Yes	Yes	No	No	Yes	Yes	No	Yes	Yes
Advanced Expression	Yes	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes	Yes
Auto Incremental	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Plug-In	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

¹ - Displays as "y"; ² - Only for find "all"; ³ - As Text

Appendix B. SmartList To Go Screen and Card Support

The SmartList To Go handheld application is compatible with the growing variety of Palm OS device screen types. SmartList To Go also includes support for Palm OS expansion cards.

In This Appendix . . .

- **B.1 Palm OS Screen Support**
- **B.4 Palm OS Expansion Card support**

B.1 Palm OS Screen Support

SmartList To Go supports a variety of device-specific features related to screen resolution, extended viewing area, and screen orientation of the device. SmartList To Go supports these features on Palm, Sony, Garmin, Handera and TapWave devices. These features are defined below:

- **Higher Resolution** – The SmartList To Go handheld application supports the higher resolution screens on applicable devices. Icons and fonts are displayed in high resolution. In addition, on high resolution devices setting throughout its functionality, including the addition of two additional icons at 32x32 and 24x24 pixels to accommodate the Handera's two additional Launcher Icon Views.
- **The Virtual Silk-Screen Area** – The SmartList To Go handheld application supports the collapsible Virtual Silk-Screen Area on applicable devices. The List View and Record screens in the SmartList To Go handheld app will now expand or collapse accordingly to take advantage of the additional display area provided by collapsing the Virtual Silk-Screen area.
- **Rotatable Screen** – The SmartList To Go handheld application supports viewing the List View and Record screens in different orientations, including both portrait and landscape orientations.

All other screens are only viewable in portrait orientation. If you open another screen while viewing a List View or Record screen in landscape orientation, the new screen will appear correctly in portrait orientation. When you return to the List View or Record screen, that screen will still be in landscape orientation.

B.2 Palm OS Expansion Card support

The SmartList To Go handheld application supports the use of Palm OS expansion card technology on Palm OS devices. A SmartList can be moved to an expansion card for storage and can be opened and used seamlessly from the card. The SmartList To Go application itself can not be stored on an expansion card and must remain on the devices main memory.

To move a SmartList to an expansion card, follow these steps:

1. With a supported expansion card inserted into the device, open the SmartList To Go handheld application.
2. Tap on the dropdown arrow to the left of the SmartList you wish to move to a card and choose **Move To Card** from the dropdown menu.
3. Notice that a card icon is displayed in the right most column. This indicates that the SmartList exists on the card.
4. To use the SmartList, simply tap on the name of the SmartList to open it.

SmartList To Go stores SmartLists on a card in a default directory called SmartList. In order for a SmartList on a card to work in SmartList To Go, it must be present in this default directory. If SmartLists exists in other locations on a card, you can search the card for those SmartLists and move them to the default directory. To search for other SmartLists, choose **Search for SmartLists...** from the main screen in SmartList To Go. If other SmartLists are found, you will be offered the option to Move these SmartLists to the default directory.

Appendix C. SmartList To Go Plug-In Support

Date Book Plug-In

1. Features Overview

The Datebook Plug-in allows the user to create appointments in their Palm Datebook application from within a SmartList record. The user can either supply all the required information to schedule the appointment, or require the user to fill in the appointment information from a popup dialog.

2. Installation

The SmartList To Go Datebook Plug-In is installed automatically when you install SmartList To Go.

3. Usage Overview

a. Setting up the Plug-in

The Datebook Plug-in uses the following fields:

- Event: A text field containing the description of the event.
- Start Time: A time field to represent the start of the appointment.
- End Time: A time field to represent the end of the appointment.
- Date: A date field to represent the date of the appointment.
- Alarm: An integer, long, or checkbox field representing the alarm setting for the appointment.
- Private: An integer, long, or checkbox field representing a private appointment.
- Note: A memo field containing a note to associate with the appointment
- Command: A constant field that can contain "Add W/ Dialog", "Add Directly", or "Remove".
- Results In: An integer, long, or checkbox value.

b. Scheduling or Deleting an Appointment

The Plug-in can perform the following commands depending upon the COMMAND field selected from the plug in properties page.

Add W/ Dialog: Use to schedule an appointment by first presenting a dialog to the user to confirm/modify the appointment details.

Add Directly: Use to schedule an appointment with the supplied parameters contained in the current record.

Remove: Use to delete an existing appointment. The plugin will delete an

existing appointment on the supplied date, with the same start and end times, and with the same description.

For all these commands, the plugin will return 1 if the action was completed successfully, or 0 if the action failed.

Timer Plug-In

1. Features Overview

The Timer PlugIn for SmartList To Go contains all the features of a standard stopwatch, allowing the user to start and stop timing, as well as pause the timer. Each timer can support a text label that can help the user keep track of which timer is currently active. The SmartList To Go Timer Plugin contains a Timer Records table, showing all the timers that are in use on the handheld device. From this screen, the user may review and/or delete timers.

2. Installation

The SmartList To Go Timer PlugIn is automatically installed during installation of SmartList To Go.

3. Usage Overview

a. Configuring PlugIn Field

From the Field Editor, select Field Type SmartList To Go PlugIn from the Field Type list. Selecting the small box to the right will open the Field Properties dialog. From the PlugIn list, select TimerPlugIn. Two fields will appear, "Text Label" and "Results In". Selecting "Text Label" will allow the user to choose a Text field to use as the timer's text label. "Results In" will allow the user to choose an Integer or Long field to store the return value (elapsed time). Both the "Text Label" and "Results In" fields are optional but recommended.

b. Starting Timer

When viewing records in Tab View, select the Timer PlugIn field to launch the timer. Note that when the timer is launched for the first time for a given record, the status label is "Not Started", the Start and Stop times are "(none)" and the Elapsed Time is 0. To start the timer, select Start Timer. The Start Time is set to the current time and the status label will change to "Running". The first button will change to "Pause Timer." Note that while the timer is running, the Elapsed Time is not dynamically updated. The Elapsed time will be updated when Pause Timer or Done Timing is selected. The initial Start Time is recorded when the timer is first started. However, the Elapsed Time reflects the total time that the timer has been in the Running state.

c. Pausing Timer

Pausing the timer will set the Stop Time to the current time and will update the Elapsed Time. To continue timing, select Resume Timer.

d. Done Timing

Done Timing, unlike Pause Timer, is not restartable. When the user returns to SmartList To Go after Done Timing, the Elapsed Time is returned, and the timer record is removed from the database. Therefore, if the user runs the Timer again after selecting "Done Timing", this has the side effect of restarting the timer in the same manner as Clear Timer (Don't press Done Timing unless you really mean it).

e. Clearing Timer

Clearing the timer will return the timer to the "Not Started" state, will set the Start and Stop times to "(none)" and will set the Elapsed Time to 0. The first button will also revert back to Start Timer.

f. Returning to SmartList To Go

The user may return to SmartList To Go no matter the state of the Timer. If timing is in progress, timing will continue even when the user has exited the Timer.

If the Timer is paused, the most recent Elapsed Time is returned. The user may run the Timer again at any time to continue timing. If Done Timing is selected, the Elapsed Time is returned, and running the Timer again will reset the timer.

4. Timer Records

a. Record Display

The Timer Records display shows a concise listing of all the timer records currently on the device. Each record contains a status (R, P, N and C for Running, Paused, Not Started, and Completed, respectively). Next, a checkbox appears. This box should be selected when deleting timer records. This is followed by a short text description, the Start Time, and the Elapsed Time. Scroll buttons appear in the bottom-right-hand corner of the screen.

b. Record Persistence

Whenever a SmartList To Go record has a timer field, a Timer Record is created. As one might imagine, this database could grow quite large if objects are never removed. To remedy this, whenever a timer is Completed (i.e. Done Timing is pressed) the Timer Record is removed from the database. All other timers maintain records in the Timer Records database.

c. Deleting Records

In the course of normal use, the user may utilize times in their databases then delete the databases, leaving the Timer Records 'stranded'. In order to give the user a way to delete those unwanted Timer Records, the Timer Records dialog displays all the timers in the handheld and allows the user to delete them. To view the Records table, select the drop-down menu and select "View Timer Records". To delete a Timer Record, select its checkbox, then select "Delete Checked Records." Note: Timer Records each take up approximately 40 bytes.

Index

Address Book 104, 110, 119
 Address Join 14, 104, 110, 119
 Add 110
 create 110
 Select 110
 use 110
 Address Lookups 14, 104, 110, 119
 Add 110
 create 110
 All 14
 Applications Launcher 11
Applying Filters 85
Applying Multiple Filters 85
 Auto Incremental 14, 119
Auto Writer 119
 Cancel button 14
Category menu 85
 Category Select 14
 Category Select menu 14
Checkbox 119
Circle Tool 99
Clear Tool 99
Cloning 84
Color Palette 99
Color Selector 99
Condition button 63
Conditional Expr 63
 Conditional Expression 14, 63, 119
Conditionial Expr 119
Configuring Filters 85
creating 110
 Address Join 110
 Address Lookup 110
Date+-Months 58
Date+-Years 58
Days 58
 DB Join 14, 104, 110, 114, 119
 create 114
 use 114
DB Join fields 114, 119
 DB Lookup 14, 104, 119
DB Lookup fields 104, 110, 119
Dec 119
Decs menu 58
Deleting Filters 85
Detail 63, 105, 114, 119
 Detail SmartList 105, 110, 114, 119
Done button 99
ELSE Response 63
Expression 58, 119
Expression Builder 58, 119
Expression Builder screen 58
Expression fields 58, 63, 119
 Field 14, 58, 63, 85, 99, 104, 105, 110, 114, 119, 124
Field Editor 58, 63, 99, 105, 110, 114, 124

Field Editor Screen 58, 63
Field Name 99, 119
Filled Circle Tool 99
Filled Rectangle Tool 99
Filter menu 85
 Filter Select 14
 Filter Select menu 14
Filters 85, 114
 Find 14
First Value 58, 63
 Form 14, 58, 105
Form Designer 58, 63
grayscale 99
Handera 330 124
 Home 11, 14
 Home icon 11, 14
IF Condition 63
Image 99
Image Color Selection 99
image icon 99
Image Information 99
Image Information button 99
Image Information Window 99
Image Screen 99
Image Tool 99
Image Tool button 99
Image Tool Size 99
Image Toolbox 99
Image Tools Size menu 99
Increment/Decrement 119
 Installing SmartList To Go 14
Jog-Wheel 124
Jog-Wheel Scrolling 124
 Join 105, 110, 114, 119
 Key 14, 105, 114, 119
 Key fields 105, 114
 Label 14, 58, 63, 105, 114
Line Tool 99
 Link icon 105, 110
List Editor 119
 List View 14, 114, 124
Logical Operator 63
 Long Integer 14, 119
 Main Menu 14, 124
Manually Recalculating Expressions 58
 Master 105, 114
 Master SmartList 105, 110, 114, 119
Microsoft Access 11
 MyBooks 14, 114
One-to-Many 104, 114, 119
 create 114
 use 114
One-to-Many button 114
One-to-Many field 104, 114
One-to-Many field button 114
One-to-Many's Filed Properties 114

Open	85
Filter	85
Palm Address Book	110
PalmOS	11, 104, 110, 119
Plug-Ins	119
Properties	58, 63, 84, 99, 105, 110, 114, 119
Properties Screen	63, 99, 105, 114
Radio Buttons	14, 119
Record 14, 58, 63, 85, 99, 104, 105, 110, 114, 119, 124	
Record Added	119
Record Browse Tool bar	14
Record Screen	14, 99, 105, 124
Record Updated	119
Rectangle Tool	99
Related SL	119
Relating	110
Relating SmartLists	104, 105
Relating SmartLists using One-to-Many	114
Relational Operator	63
Response	63
Second Value	58, 63
SmartList	58, 94, 104, 105, 110, 114, 119, 124
beam	94
clone	94
Closing	94
viewing	94
SmartList To Go 4, 11, 12, 13, 17, 18, 19, 20, 21, 22, 23, 24, 26, 28, 29, 31, 32, 35, 36, 40, 43, 44, 45, 52, 53, 54, 58, 61, 68, 70, 71, 72, 76, 77, 78,	

79, 80, 81, 82, 83, 84, 85, 86, 88, 89, 90, 94, 99, 104, 114, 115, 117, 119, 121, 123, 124	
SmartList To Go Home	11, 124
SmartList To Go Home Screen	124
SmartList To Go Plug-In	119
SmartLists	104, 105, 114, 124
Source SL	105, 119
Support	123
Text	14, 63, 99, 110, 114, 119
Text Tool	99
THEN	63
THEN Response	63
This SL	114
This SL menu	114
Toolbar	99
Type	13, 14, 58, 63, 99, 104, 105, 110, 114, 119
use	114
DB Join	114
One-to-Many	114
Using Filters	85
Verify Password window	94
View Landscape	124
View List	14, 114
View Select	14
viewing	94
SmartList	94
Viewing Area	99
Views	14, 85, 99, 105, 110, 114, 119
Virtual Silk-Screen Area	124