

V I S T A

File Manager 22.2
Getting Started Manual
March 2013



Fileman/
Lab
Agile
Project



©Copyright 2013 by VISTA Expertise Network. Licensed under Creative Commons Attribution-ShareAlike 3.0. Details are available at <http://creativecommons.org/licenses/by-sa/3.0/>.

Revision History

Date	Description	Language	Authors
January 2013	Version 22.2 release	English (US)	Kathy Ice and Frederick D.S. Marshall, based on the work of Tami Winn, Michael Ogi, Thom Blom, Susan Strack, Jack Schram, and Gary Beuschel

Contents

- Orientation.....1
- Getting Help.....2
- Introduction.....3
- What is Fileman?.....3
- Before You Start.....3
- Part 1: End Users.....5**
- Chapter 1: Interacting with Fileman.....5
- Your User Interface.....5
- Responding to Prompts.....6
- Chapter 2: Field Types.....9
- Using VISTA's Word-Processing Tools.....12
- Chapter 3: Getting Fancy.....16
- Partial Responses.....16
- Longer Default Responses.....17
- Jumping Around.....19
- Required Fields, Required Identifiers, and Key Fields.....21
- More About Field Prompts.....21
- Typical Data Entry Session.....22
- Your Next Steps.....23
- Part 2: Super-Users.....25**
- Chapter 4: Going Backstage.....25
- Super-User Access.....25
- Chapter 5: Inquire.....28
- Inquire to File Entries Option.....28
- Chapter 6: Print or Display.....31
- A Word About Printing.....31
- Printing a Default Report.....31
- Custom Reports: Sorting.....33
- Custom Reports: Choosing Print Fields.....39

Custom Reports: Final Touches.....	41
Chapter 7: More Inquire and Print Options.....	46
Computed Fields.....	47
Choosing a Device.....	47
More Sorting Options.....	48
More Printing Options.....	49
Editing Templates.....	51
Sharing Templates.....	54
Chapter 8: Search.....	56
How Searching Works.....	56
Entering Conditions.....	56
Combining Conditions.....	58
Search Templates.....	62
Formatting Output.....	63
Using Your Templates.....	64
Searching Multiples.....	65
Chapter 9: Viewing Reports on the Browser.....	67
The Browser Screen.....	67
Browser Features.....	68
Chapter 10: Editing Records.....	71
The .01 Field.....	71
Enter or Edit File Entries Option.....	71
Using ^LOOP.....	72
How to Stuff a Field.....	74
Using Templates.....	75
Edit Qualifiers.....	76
Your Next Steps.....	80
Appendix A: Sort and Print Qualifiers.....	81
Sort Qualifier Reference.....	81
Print Qualifier Reference.....	84
Glossary.....	89
Index.....	95

Orientation

As the name implies, this manual shows first-time users some basic features and options that will get them started using Fileman 22.2. The manual assumes that Fileman 22.2 is already installed and running on your system.

This manual is not intended to be a comprehensive list of all features available in Fileman, or even all features available to users and super-users. Rather, it is intended to walk new users through the most commonly-used features and give them an idea of where to go next for more information.

This manual uses VA conventions for displaying mockups of sensitive data in illustrations and screen captures. The first three digits (prefix) of any social security numbers (SSN) will begin with either “000” or “666.” Patient and user names will be formatted as FMPATIENT,[N] and FMUSER,[N] respectively, where “N” represents the first name as a number spelled out.

Screen captures and computer source code are shown in a non-proportional font and enclosed within a box. The user's responses to online prompts will be boldface.

Most screen captures are shown in scrolling mode. This is because scrolling mode prompts and responses are easier to follow on paper or in PDF form. Most of what is shown in scrolling mode can also be done in screen mode, list mode, or with a GUI; it will just look a little different.

This manual is in two sections, because it is intended for two slightly different audiences. The first section contains information for end users. The second section contains information for “super-users,” a general term that encompasses Application Coordinators, CACs, ADPACs, and those with similar roles.

Getting Help

Fileman 22.2 includes online help that is available to you while you are in VISTA. In general, whenever you are at a prompt, you can type one or two question marks for additional information.

If you need additional help with Fileman, contact your supervisor, your Application Coordinator, or the IT representative for your service. If those are not good resources, check with the OSEHRA community at www.osehra.org. You should be able to find help from more experienced users or programmers.

A more in-depth discussion of the basics of VISTA's menu and data-entry systems is available in *Getting Started in VISTA*, a book available from www.vistaexpertise.net.

Introduction

What is Fileman?

In discussing what Fileman *is*, it is probably best to begin with what Fileman *isn't*. Fileman isn't a clinical package like Laboratory or Pharmacy, and it isn't an administrative package like IFCAP (Integrated Funds Distribution, Control Point Activity, Accounting, and Procurement). Fileman is an infrastructure package, meaning it is part of what makes VISTA run. Programmers use the features in Fileman when creating those other packages: Lab and Pharmacy and IFCAP and so on. Because the same infrastructure packages and tools are used throughout VISTA, similar commands, keystrokes, and shortcuts can be used in all of them. In learning to use the features in Fileman, you are learning to use features in all of the other packages.

“Fileman” is the shorter, more informal name of the package; File Manager is its formal name. As you may have guessed from the name, the specific part of the infrastructure that File Manager handles is files. That is, Fileman manages VISTA's massive database. Fileman commands are all about data: entering it, editing it, retrieving it, sorting it, printing it, making it into reports.

Although Fileman was written for VISTA, it does not need to be used with VISTA. It can be installed as a standalone database management system.

Before You Start

In order to use Fileman, you should have a basic working understanding of VISTA, including how to sign on, how to navigate using menus, and how to access help by typing one or two question marks. Because Fileman is a database system, you should also have a basic understanding of what a

database is, and the meaning of basic database elements such as field, record, and file.

Before starting on the second section of this book, you should make sure that you are reasonably familiar with the material in the first section. It is not necessary that you already have a “super-user” job title, just that you are interested in learning the material.

Part 1: End Users

Chapter 1: Interacting with Fileman

As an end user, you do not interact directly with Fileman. Since it is an infrastructure package, the only ones who deal directly with Fileman are super-users and programmers.

So, why learn about it in the first place? As we said in the Introduction, Fileman underlies all of the packages you do use. Learning the basics of Fileman means getting a head start on all the packages you need to use for your job, and any packages you may need to learn in the future.

Your User Interface

In VISTA, there are three native methods for displaying information and prompts to the user: scrolling mode, screen mode, and list mode. In addition, development teams for some packages have added a Graphical User Interface (GUI) for a windows-like look and feel.

In scrolling mode (often called “roll-and-scroll”), you will respond to one prompt at a time. The prompts will be given to you in a pre-determined order, designed by the package’s development team. You can receive help at any prompt by typing one or two question marks.

In screen mode, you will see an entire screen “form”, which you can navigate through using command keys. Typing one or two question marks will give you more information about how to navigate and use screen forms.

The third kind of display, list mode, is less common. In list mode, you see a list of your data, and an action area in which you can type commands. As

with the other modes, you can get help in list mode by typing one or two question marks.

GUIs are popular with users, but it's important to remember that they are not part of Fileman. They are written in other programming languages and added on. Getting help in a GUI-based application will depend on how the GUI has been set up. The documentation for the specific package should include information on how to get help.

The environment used by a specific package—scrolling mode, screen mode, list mode, or GUI—is determined by the package's development team. You, as the user, cannot switch to a different mode. You should therefore become familiar with scrolling mode and screen mode (and list mode, if applicable), so that you can use all the applications your job requires.

Responding to Prompts

As a user, your interactions with Fileman will consist mainly of responding to prompts. The prompts will ask you to enter information, or choose information from a list. When you respond to a prompt, pressing the Enter or Return key on your keyboard is how you tell Fileman that your answer is complete. If you type an answer, or choose an item, and do not press Return, Fileman will think you're still working on it and will continue waiting patiently for you.

Basic Prompts

A basic Fileman prompt looks like this:

COUNTY :

With a prompt like this, Fileman is asking for information. Usually, when Fileman is expecting information, there will be a colon at the end of the text; the colon is your clue that you need to enter something. In this case, Fileman is looking for the name of a county.

Many Fileman prompts look more like this:

```
COUNTY: ALAMEDA//
```

With this prompt, a default answer (also called a default) is supplied by Fileman. You can easily see that it is a default because it is followed by two slash marks. Defaults work a little differently for different kinds of prompts; sometimes you'll see the most common answer, sometimes you'll see the most recent answer, sometimes you'll see something else.

To accept the default, press Return. To enter a different answer, just start typing. Your answer will appear immediately after the slash marks, like this:

```
COUNTY: ALAMEDA// SANTA CLARA
```

When you press Return after typing your answer, Fileman checks to make sure your answer fits the kind of answer expected. If it does, your answer is accepted. Whether this also changes the default answer will depend on how the programmer set up this particular prompt.

Select Prompts

When a prompt begins with the word "Select," Fileman is expecting you to choose an answer from an existing list. Select prompts almost always have default answers.

```
Select STATE: CALIFORNIA//
```

Because select prompts are expecting a very specific response, it's not a bad idea to type a question mark so you can get a better idea of what Fileman is looking for.

Incorrect Responses

If you respond to a prompt (whether a basic prompt or a select prompt)

with an incorrect response, the world will end, and all life as we know it will cease to exist.

Wait, that's wrong. All that happens is that the computer alerts you, and asks you to try again. The world doesn't end, and life goes on.

What constitutes an incorrect response? That depends on the situation. You may have typed in a word where Fileman was expecting a number. You may have entered a social security number without dashes, when Fileman was expecting dashes. Fileman has some pretty clever ways of checking input to make sure it's the kind of input expected. A big part of that checking has to do with field types, which we discuss in the next chapter.

Skipping a Prompt

If a prompt does not have a default answer, you can skip it by pressing Return without entering any information. This tells Fileman to leave that field blank and continue on to the next prompt.

Remember that if a prompt does have a default, then pressing Return without entering information is how you accept the default. Pressing Return in this case is not really "skipping" the prompt; it's accepting the answer shown.

Chapter 2: Field Types

When programmers set up a field in Fileman, they designate it as one of nine field types. These types are:

- Computed
- MUMPS
- Date
- Free Text
- Numeric
- Pointer to a File
- Set of Codes
- Variable Pointer to Files
- Word-Processing

Of these types, there are two where end users generally do not enter or edit information. Computed fields, as the name suggests, are derived from the values of other fields; they are almost always locked to prevent changes. MUMPS fields are where programmers can enter commands in the MUMPS programming language—something you will not be called on to do.

Date fields are designed to hold calendar dates. You can enter dates in several formats; in fact, Fileman 22.2 has added even more formats that will be accepted. Type a question mark at the prompt to see all the formats available.

Some date fields will accept a partial date: the month and year only, or just the year. Again, typing one question mark at the prompt will show you your options.

Fileman is pretty smart. If you use a two-digit number for the year, Fileman interprets it to mean a future date if the date would be within twenty years from now. For example, if you typed “24” for the year, Fileman would interpret that as 2024. If it would be more than 20 years, Fileman assumes

the date is from the past. For example, if you typed “54,” Fileman would interpret that as 1954.

In addition, Fileman provides some handy tricks and shortcuts for entering dates. You can enter the word TODAY, or just the letter T, and Fileman will automatically put in today’s date. You can add or subtract days, weeks, or months to TODAY, and Fileman will do those calculations as well.

Examples:

T+3D	Three days from today
T+3	Three days from today (Day is the default)
T+3W	Three weeks from today
T-1M	One month ago
T-2D	The day before yesterday

Some date fields are really date/time fields; they will accept a time as well as a date. As with date-only fields, typing a question mark at the prompt will give you an idea of what formats you can use at a date/time prompt.

Fileman is pretty clever about times, too. If you enter a time without specifying AM or PM, Fileman interprets it to be during the day shift. For example, 10:15 would be interpreted as 10:15 AM, but 3:30 would be interpreted as 3:30 PM.

As with dates, Fileman has some tricks and shortcuts for entering times. You can use MID for midnight and NOON for noon. You can enter the word NOW, or just the letter N, and Fileman will enter the current time. You can add or subtract hours or minutes to NOW, as in these examples:

N+1H	One hour from now
N-30'	Thirty minutes ago
N+4H	Four hours from now

Free text fields, as the name implies, will accept just about any input. They may have minimum or maximum lengths, which you can determine by typing a question mark at the prompt.

There are a few characters that free text fields will not accept. Fileman usually won't accept the up-arrow or caret (^), because it is often used as part of VISTA commands. And although Fileman will accept an @ as part of an answer (in an email address, for example), it will usually not accept an @ by itself. It's also a bit of a challenge to enter a question mark by itself. Fileman will think you're asking for help!

Numeric fields accept only numbers. If you try to type in a word, your answer will be rejected. Numeric fields are often set up to accept only numbers within a certain range. You can type a single question mark at the prompt to find out what kind of number Fileman is expecting.

Pointer fields "point" to data that resides in another file. For example, when you are asked to enter an address, the field for "state" is probably a pointer to the STATE file. Information about states lives in the STATE file, but you can use it for the address you're entering. Responding to a pointer prompt is a lot like responding to a select prompt, except you won't see the word "select." Also, you normally won't be able to change the data in a pointer field, since the actual data resides someplace else.

Some fields are expecting an entry from a specific **set of codes**. Examples include M or F for gender, or Y or N for yes or no. In this kind of field, Fileman will accept only one of the codes specified; no other entry will be accepted. Typing a single question mark at a set-of-codes prompt will give you information about how to answer.

Variable pointers work exactly like pointer fields, except they can point to several different files to access their information. As with pointer fields, answering these fields is a lot like answering select prompts, and you normally won't be able to change the information.

Word-processing fields are similar to free-text fields in that they will accept just about any combination of letters, numbers, and symbols. However, word-processing fields typically allow longer responses. More importantly, when you reach a word-processing field, Fileman launches one of VISTA's

native word-processing tools so that you can compose your answer. You can learn more about the word-processing tools in the next section.

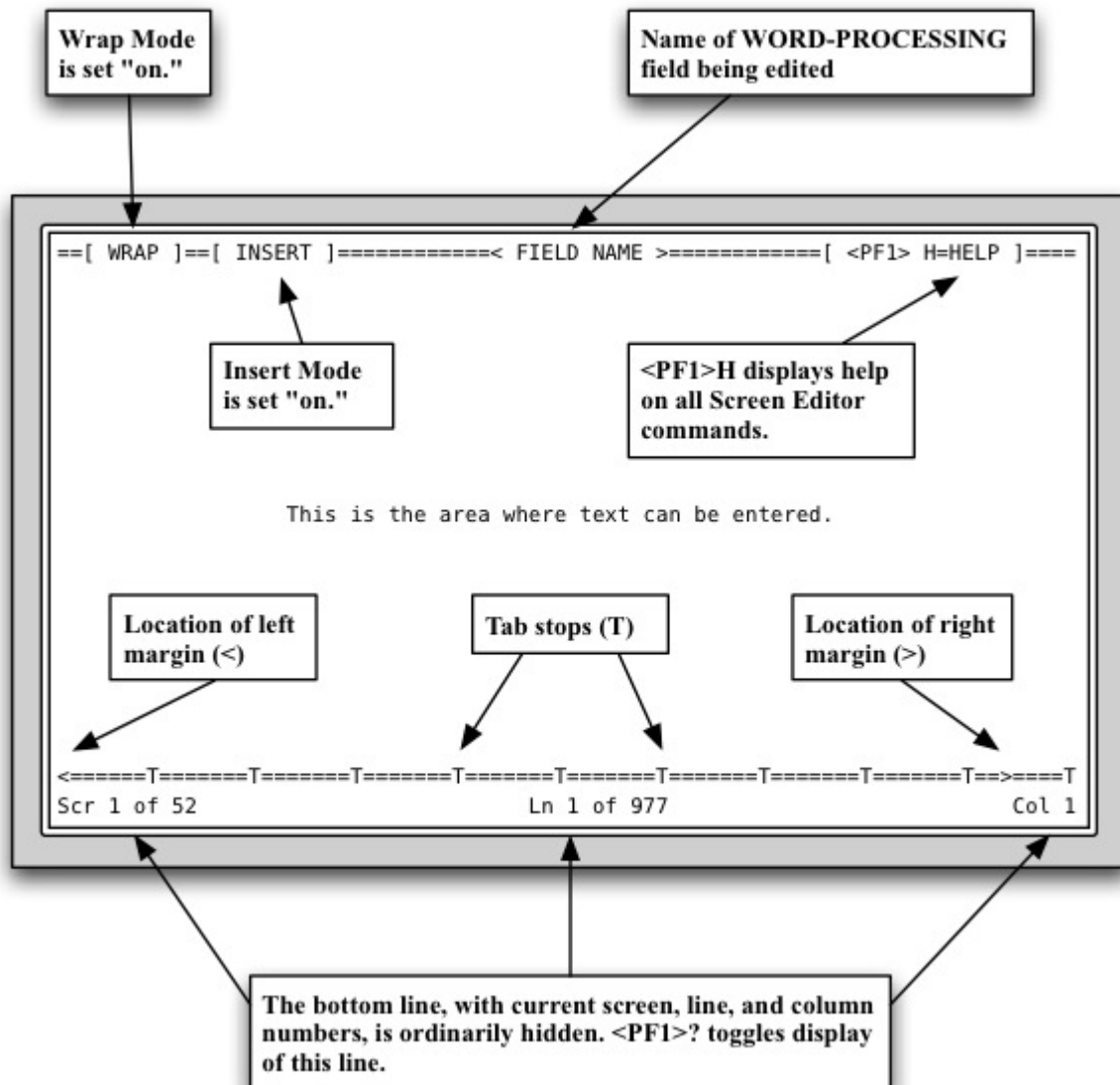
Using VISTA's Word-Processing Tools

VISTA has two native tools for answering a word-processing field: the screen editor and the line editor. For users accustomed to commercial word-processing software such as Microsoft Word, these editors may come as something of a surprise. There is no point-and-click, no handy formatting buttons, no pull-down menus. However, both tools do a lot of the same things the commercial packages do; it's simply a matter of learning how to make them do it.

Of the two tools, the screen editor is probably the best option for users accustomed to modern word processors. We recommend setting the screen editor as your preferred word-processing tool. To do this, enter **TBOX** at any menu prompt. This should take you into the User's Toolbox menu. From there, select **Edit User Characteristics**, then go to the Preferred Editor field and make sure it is set to screen editor.

Using the Screen Editor

The screen editor, as the name implies, provides you with a whole screen on which you can edit your text. The figure below illustrates a blank word-processing screen. If you immediately begin typing, your response will appear in the large area in the center of the screen.



All commands in the screen editor begin by pressing the <PF1> key. Pressing this key tells the screen editor that what comes next is a command, and not a letter to be typed into the response. The odds are pretty good that your computer doesn't actually have a <PF1> key on its keyboard, but don't panic! Your IT department has almost certainly mapped the <PF1> key to a different key, or combination of keys, that you can press instead. What has it been mapped to? We don't know; there are several options, and it would have been done locally, there at your facility. If your keyboard has an <F1> key, you can try that and see what happens, but you will probably

need to ask your supervisor or Application Coordinator what the proper sequence is.

If you press **<PF1>** and then **H** (for Help), you will see a list of the most commonly-used commands in the screen editor.

The main thing to remember in the screen editor is that it isn't a windows-type application. You will not be able to use your mouse. You can navigate using your arrow keys, your tab key, and special commands such as **<PF1>T** for top, and **<PF1>B** for bottom.

Notes on the Line Editor

The line editor is older than the screen editor, and more challenging for modern users. We recommend using the screen editor instead, and if you have set your User Characteristics to use the screen editor, you should never encounter the line editor.

If you do encounter the line editor, you will see the name of the field you are working in, followed by a numbered line, like this:

```
Description:  
1>
```

If you're feeling brave, or if your reply is short, you can just start typing. However, we recommend switching to the screen editor. To do this, press the **Return** key without entering any text. That should give you a prompt that looks like this:

```
EDIT Option:
```

At this prompt, type a **U** to reach the Utility submenu. The first option on this submenu allows you to exit the line editor and continue the task in the screen editor.

After completing your task, we recommend returning to your User's Toolbox and the Edit User Characteristics option, and making sure your

preference is set to use the screen editor. The line editor is interesting, and you are welcome to learn to use it if you want to, but you should be able to complete all your word-processing tasks with the screen editor.

Your Alternate Editor

As we mentioned earlier, the screen editor and the line editor are the word-processing tools native to VISTA. However, your IT department may have provided you with an alternate editor for word processing. How can you tell if you have an alternate editor? If you type **TBOX** at any prompt, then select **Edit User Characteristics**, and look at the Preferred Editor field, you should see your alternate editor, if you have one. Everyone should have the line editor and the screen editor; if you have a third one listed, that is your alternate editor.

If you do have an alternate editor, we can't tell you what it's called or what it does, because alternate editors vary from site to site. Contact your supervisor or IT department for information about your alternate editor. If you like your alternate editor, you can choose it for your preferred editor and use it all the time!

Chapter 3: Getting Fancy

The first two chapters provide some basic information about how to use Fileman. Now that you know the basics, there are some nifty tricks you can use to save time when working in Fileman.

Partial Responses

Raise your hand if you want to type ACETAMINOPHEN twenty times a day. We didn't think so. When choosing an answer from a select prompt, you can just type the first few letters of your answer before pressing Return. This is called a partial response.

If there is only one item on the list that matches your partial response, Fileman automatically enters it as your answer. If there is more than one possibility, Fileman gives you a numbered list. You can then select your answer using its number.

Keep in mind that a partial response only works if the answer is already in the database. Fileman is smart, but it doesn't read minds. If your answer isn't in the database, you'll need to add it by typing the entire thing. (If you find you need to add an answer and are not allowed to, contact your supervisor or Application Coordinator.)

Multi-Part Responses

Sometimes, just typing the first part of the first word of the answer you want to select isn't going to do the trick. With some select prompts, you can use a multi-part response, which uses a comma to separate words—or parts of words—that you are looking for. For example, you might be trying to look up a doctor. You remember that his first name is Tim or Tom, and his last name is Brad-something. Bradford? Bradshaw? Bradley? Something like that. You can use a multi-part response to find the doctor based on this partial information:

```
Select DOCTOR: T,BRAD <Return>
```

- 1 T. Bradford Griffin
- 2 Tamara Bradlee
- 3 Thomas Bradwell

```
Choose 1-3:
```

Fileman responds with a list of entries that match your multi-part response.

Note that multi-part responses are not as flexible as, say, Google searches. Each part of your multi-part response has to be the *beginning* of the word; you cannot search for strings found within the word. In addition, the parts have to be in the order you typed them. Our T,BRAD search, for example, would not find a doctor named Bradley Thompson because the words are in a different order.

Longer Default Responses

When a default is 20 or more characters in length, it won't be followed by double slashes (/ /). Instead, it is presented with a "Replace" prompt:

```
ADDRESS 1: 301 Bayshore Boulevard  Replace
```

Accepting this kind of default is exactly the same as accepting shorter default answers; you simply press the Return key.

If the answer you want is kind of similar to the default, but something needs to be changed, you can tell Fileman which parts need to be replaced. Fileman will respond with a "With" prompt, allowing you to type in the replacement text. Here is an example:

```
ADDRESS 1: 301 Bayshore Boulevard  Replace Bay <Enter> With
North <Enter>
Replace
```

In this example, we're asking Fileman to replace the word Bay with the word North. Fileman responds with another Replace prompt, so we could

change more about the original address if we wanted. Otherwise, we can press **Return** without entering any text to signal to Fileman that we're done replacing. The whole dialog would then look like this:

```
ADDRESS 1: 301 Bayshore Boulevard  Replace Bay <Enter> With
North <Enter>
Replace <Enter>
    301 Northshore Boulevard
ADDRESS 2:
```

So, if the address you want looks nothing like the default address, do you have to type the entire default in order to replace it? You do not. You can enter ellipses (three periods) to indicate that you would like to replace the entire answer.

```
ADDRESS 1: 301 Bayshore Boulevard  Replace ... <Enter> With 5995
Folsom Street <Enter>
Replace <Enter>
    5995 Folsom Street
ADDRESS 2:
```

More About Ellipses

In the above example, we showed you how to use ellipses (...) to replace an entire answer. You can also use ellipses to replace part of an answer.

At the beginning of an answer:

```
ADDRESS 1: 301 Bayshore Boulevard  Replace ...Bay <Enter> With
499 North <Enter>
Replace <Enter>
    499 Northshore Boulevard
ADDRESS 2:
```

In the middle of an answer:

```
ADDRESS 1: 301 Bayshore Boulevard  Replace 1...ore <Enter> With
15 Lakeside <Enter>
Replace <Enter>
    3015 Lakeside Boulevard
ADDRESS 2:
```

At the end of an answer:

```
ADDRESS 1: 301 Bayshore Boulevard  Replace Bou... <Enter> With
Drive <Enter>
Replace <Enter>
      301 Bayshore Drive
ADDRESS 2:
```

Caution!

“Replace...With” is a useful tool, but you need to pay attention when using it. Fileman will only replace the *first instance* of a sequence in the answer. Consider this example:

```
ADDRESS 1: 1026 Eastlake Avenue East  Replace East <Enter> With
West <Enter>
Replace <Enter>
1026 Westlake Avenue East
      ADDRESS 2:
```

Now, if you *meant* to change the address to Westlake Avenue East, that’s great. If you didn’t...well, you messed up. It’s easy to mess up if you go too fast and don’t think your answer through.

Keep in mind, however, that messing up is not the end of the world. You can fix your answer and keep going—as long as you’re paying attention and noticing what Fileman does with your answers.

Jumping Around

When you’re in scrolling mode, fields for information are presented to you one at a time. In screen mode, your cursor advances through the fields in a predetermined order. We have already discussed how you can skip fields by pressing Return without entering information.

However, you can also jump from the current field to another field—either an earlier field you already finished, or a later field you need to get to. To jump, enter an up-arrow or caret (^), and then the first few letters of the

field you want. In the following example, the user jumps from the SSN field to the date of birth field.

```
SSN: 000123123// ^DATE OF BIRTH <Enter>
DATE OF BIRTH:
```

To get a list of the fields to which you can jump, enter a caret and a question mark (^?) at any field prompt.

Jumping Out

You can also use your up-arrow to jump out of a record altogether. For example, let's say you selected a patient record to edit, then realized you had the wrong patient. You can type an up-arrow, by itself, at any field prompt to jump out of this patient's record and select another one.

```
SSN: 000123123// ^ <Enter>
Select PATIENT NAME:
```

Entering the up-arrow by itself to exit a given function is a convention used throughout VISTA applications.

Cancel (^)

Usually, when you ask VISTA (and, by extension Fileman) to look up or retrieve information for you, you get a response almost instantly. If traffic is heavy, it may take as long as a second.

If it seems that a search is taking too long, it may be an indication that there's a problem with the system (or that you accidentally gave Fileman a search that's really complex). If it seems you've been waiting too long for a reply, you can type two up-arrows (^) and then press the Return key, to tell Fileman to cancel the search.

Required Fields, Required Identifiers, and Key Fields

Some Fileman fields have been set up as required fields. You will be able to tell when you reach one of these fields, because you will be unable to skip it by pressing the Return key. However, you can jump out using an up-arrow.

Some required fields are also required identifiers. This means that if the field contains no information, the record will not be saved. For example, if you are trying to create a new record that is a patient appointment, the date field might be a required identifier. If you left the date field blank, Fileman would not save the appointment. You would receive a warning before the appointment was discarded, so you could correct the problem.

In a key field, the information is not only required, it must be unique. For example, each patient should have a unique Patient ID number; if you try to enter two patients with the same Patient ID, Fileman will alert you, and will not accept the duplicate information.

More About Field Prompts

Deleting a Field's Value (@)

We have already showed you how to accept a default response, and how to enter a different response. But if a field has a default, how do you tell Fileman to leave it blank? You can do this by typing an @ at the prompt, as in the example below:

```
DATE OF BIRTH:  May 21, 1946// @ <Enter>
SURE YOU WANT TO DELETE?
```

In this example, the date on file would be erased: there is no answer to the DATE OF BIRTH question; its value is now null. You're asked to confirm the deletion; this gives you a chance to change your mind before deleting the field.

In chapter 2, we mentioned that a free-text field will not accept an @ by itself. This is one of the reasons; an @ by itself means “delete the information in this field and leave it blank.”

Spacebar Recall

This is a neat trick you can do with all select prompts. If you want the same answer you picked the last time you saw this select prompt, press the spacebar and then the Return key. Your previous answer will be selected. For example, if you are editing a particular patient in a Nursing application and then switch to the Order/Entry application to work with the same patient, you can use the spacebar to select the same patient at the Order/Entry “Select PATIENT NAME:” prompt:

```
Select PATIENT NAME: <Space><Enter>
      FMPATIENT,TWENTY-FOUR
NAME: FMPATIENT,TWENTY-FOUR//
```

Spacebar recall should work at any select prompt. It may also work at other prompts; you can try it and see what happens.

Here is a secret: VISTA programmers can add spacebar recall to just about any prompt you encounter. If you find yourself really wishing that a particular prompt had a spacebar recall, you can ask for it. Contact your IT department and see what they say!

Typical Data Entry Session

So far we have discussed responding to individual prompts. Here is an example of a typical data entry session. The example is characteristic of editing data in many VISTA applications:

```
Select Patient (Name or SSN): FMPATIENT, <Enter> 25      01-12-41
000456789      COLLATERAL

Height: 5' 4"// <Enter>
Weight: 150#
```

```
Date Weight Taken: TODAY// <Enter> (MAY 17, 2013)
Usual Weight: 145#
Wrist Circumference (cm): <Enter>
Frame Size (SMALL,MEDIUM,LARGE) MED// <Enter>

Calculation of Ideal Body Weight
  H Hamwi
  M Metropolitan 83
  S Spinal Cord Injury
  E Enter Manually

Method: S <Enter>

Extent of Injury:
  P Paraplegic
  Q Quadriplegic

Select: P <Enter>

Select Ideal Weight (109-118) 114 lb // <Enter>
Does Patient have an Amputation? NO// <Enter>
Do you wish Anthropometric Assessment? NO// <Enter>

Collecting laboratory data ...

Calculate Energy Requirements Based On:
  1 Actual Body Weight
  2 Ideal Body Weight
  3 Obese Calculation

Choose: 1 <Enter>

Comments:
  No existing text
  Edit? NO// <Enter>

  Do you wish to FILE this Assessment Y// <Enter>
```

Your Next Steps

You now know just about everything end users need to know about Fileman. As we've said, most of your interactions with Fileman will take place through one of the packages you use for your job. A good next step

would be to find the user manual or getting-started manual for those packages, and begin familiarizing yourself with how they work.

Of course, if you're curious and want to learn more about Fileman, you certainly can. In that case, a good next step would be to read the next part of this manual, about how super-users interact with Fileman.

Part 2: Super-Users

Chapter 4: Going Backstage

As we said in Part 1, end users only interact with Fileman through the various packages that use Fileman's features. The fields they respond to, and the reports they can see, are all created by package developers using Fileman's tools. As a super-user, you can go "backstage" in Fileman, to see how things work behind the scenes. As is typical of backstage areas, basic Fileman is pretty utilitarian. The pretty backdrops are facing the audience; back here there are unpainted two-by-fours and exposed nails and a few wires to trip over.

If you know what you're doing, however, Fileman can help you retrieve data in combinations not available to users. You can search for specific kinds of data, and create and save your own reports. You can do data cleanup to fix mistakes that have made their way into the database. You can help your site's programmers custom-tailor VISTA to better suit your facility and your department.

Super-User Access

"Super-user" is a catch-all term encompassing CACs, ADPACs, Application Coordinators, and similar titles. Different facilities use different terminology to describe their super-users, but basically, super-users are the ones who manage VISTA for their service area. If you are interested in learning more about Fileman, and your job title doesn't (yet!) designate you as a super-user, you are still welcome to read this section.

Super users can be given access to Fileman features and options that are not available to regular users. Your supervisor or IT department can help

you get this access, if you don't have it already. Getting super-user access to Fileman, however, is only the first step.

Fileman, as the name suggests, manages files. In order to use Fileman's super-user features, you need to have access to at least one of these files. Each file in your facility's VISTA system has its own separate security, so access can be given one file at a time.

If you use a super-user option, and ask for a file you don't have access to, Fileman will react as though that file does not exist. It won't give you a message like "you don't have permission to access that file;" it will just give you two question marks, as if it has never even heard of the file.

This is actually on purpose, for security reasons. A hacker would not be able to figure out which files a system has by trying some likely-sounding file names, and seeing which ones are "not found" and which ones are "you don't have access." If Fileman reacts exactly the same way to both scenarios, the system is more secure.

Of course, it's a little disconcerting to ask for a file you know perfectly well is in there, only to have Fileman tell you it isn't. Don't panic! You haven't broken Fileman; you just don't have permission to access that file. Talk to your supervisor or IT department to get the access you need.

There are four levels of access to Fileman files:

Read access: you can see what's in the files, but you can't add or change anything.

Write access: you can edit records, but you can't add or delete them.

LAYGO access: you can add, but not delete records.

Delete access: you can delete records.

Delete access is the highest security level, because deleting a record is one of the the most drastic things you can do in a database. Incorrect changes can be fixed, but it's very difficult to undo a whole-record deletion.

When you speak to your supervisor or IT department about which files you need access to, it is important to tell them what kind of access you want. Typically, a super-user will have Read access to several files, and Write access to only a few files. They may not have any LAYGO or Delete access at all; if they do, it's typically only one or two files.

Before you start composing a list of files you need access to, you should know that programmers can copy another user's access profile, and apply it to you. If, for example, you are replacing someone else as a super-user, you can just ask your IT department to give you the same access to the same files that the other person had. That can save time—both for you, and for the programmers.

If you are not yet officially a super-user, you may still be able to get super-user access, and access to a demo file so you can practice without actually affecting your facility's real database. It might not be allowed, but it won't hurt to ask your IT department about it.

Chapter 5: Inquire

End users can look at Fileman data by using the various reports supplied by the package developers for the particular package that they are using. As a super-user, you have other ways of retrieving and reviewing Fileman data.

Inquire to File Entries Option

The most basic of these options is called Inquire to File Entries. You can use this option when you know the name of the file you want to look at, and you know specifically which records you want to see.

To use this option, type **INQUIRE TO FILE ENTRIES** (or just **INQ**) at your menu prompt. Fileman responds by asking you what file. For our examples, we'll use an imaginary file called **COMPOSER**.

```
Select OPTION: INQ <Enter>
OUTPUT FROM WHAT FILE: COMPOSER// <Enter>
Select COMPOSER NAME:
```

Now that you have the file, Fileman asks which specific records you want to look at. We need to respond with (in this case) the specific names of the composers we want to see.

This is Fileman's simplest type of database query, because it only shows you what you specifically ask for. Using this option, we could not (for example) ask Fileman to display all composers born before 1800. That's a more advanced type of query we'll learn about in the next chapter. For now, we can just request, and see, specific records in this file. So, let's start requesting.

```
Select OPTION: INQ <Enter>
OUTPUT FROM WHAT FILE: COMPOSER// <Enter>
```



```
Select COMPOSER NAME: MOZART <Enter>
```

When we type the name of our first composer, Fileman finds the record and echoes back the full name, plus any additional identifiers the record may have.

```
Select COMPOSER NAME: MOZART, WOLFGANG AMADEUS
Another one: HAYDN, JOSEPH
Another one: BACH <Enter>
  1 BACH, CARL PHILIPP EMANUEL
  2 BACH, JOHANN CHRISTIAN
  3 BACH, JOHANN SEBASTIAN
Choose 1-3: 3 BACH, JOHANN SEBASTIAN
Another one: <Enter>
```

After we select the first record, Fileman prompts us for “another one.” If we type in HAYDN, and press the return key, Fileman responds with the full name. As with select prompts elsewhere in VISTA, if there is more than one record that matches what we type in, Fileman asks us to choose one record from a list. In this example, typing BACH yielded a list of three possible Bachs. When we choose number 3, Fileman fills in the full name.

When we’re done choosing our composers, we press the return key without entering information. This signals to Fileman that we’re done picking records, and we’re ready to move on to the next steps.

```
Standard Captioned Output? Yes// <Enter>
Include COMPUTED fields: (N/Y/R/B) No// <Enter>
Display audit trail? No// <Enter>
```

What are we being asked here? The first question is asking whether we want to accept Fileman’s default format for displaying our composers. In standard captioned output, all fields that contain data are displayed on your computer in a format determined by Fileman. If you answer “No” to this question, you can choose which fields to show, how to arrange them, and whether to display them on your screen or print them on a printer. For now, though, we’re just going to accept the default.

Next, Fileman asks if we want to include computed fields. There are several

options here, which we will discuss later, but for now we'll just accept the default answer of "No."

Depending on the security settings of the file, Fileman may ask you if you want to display an audit trail. This feature is useful for security and patient-privacy officers in your facility, but it's not all that useful for a standard super-user. If you see this prompt (and you probably won't), you can accept the default answer of "No."

Fileman then displays the results of our inquiry, which look like this:

NAME: BACH, JOHANN SEBASTIAN	BIRTHDATE: MARCH 31, 1685
BIRTHPLACE: EISENACH	ERA: BAROQUE
DIED: JULY 28, 1750	
NAME: HAYDN, JOSEPH	BIRTHDATE: MARCH 31, 1732
BIRTHPLACE: ROHRAU	DIED: MAY 31, 1809
NAME: MOZART, WOLFGANG AMADEUS	BIRTHDATE: JANUARY 27, 1756
BIRTHPLACE: SALZBURG	ERA: CLASSICAL
DIED: DECEMBER 5, 1791	

Notice that in the record for Joseph Haydn, there is no entry for the ERA field. Fileman simply skips this field and moves on to the next.

In Chapter 7, we'll learn more about Inquire to File Entries, and how we can customize the output. For now, though, let's leave inquiry for the time being and move on to printing and displaying reports.

Chapter 6: Print or Display

A Word About Printing

As you have probably noticed, VISTA and Fileman are rather old. This is most obvious in the user interface, which often seems old-fashioned, but it occasionally shows up in terminology as well. For example, some VISTA documentation still refers to your computer as your “terminal.”

“Print to screen” is another piece of old computing terminology that may be unfamiliar. On modern computers, printing information and showing information on the computer screen are two completely different ideas. In older systems, though, “print to screen,” was a way of telling the system, “take this information, arrange it as if you were going to print it, and then show it to me on my computer.” Or rather, “on my terminal.”

We mention this because in many VISTA applications, including Fileman, you often have to select a “print” option in order to display information on your screen the way you want to see it. It’s a little counter-intuitive at first, but keep in mind that VISTA can output information to any device. Yes, the “device” is usually a printer, but it can treat your screen as a “device” as well.

Therefore, in this manual (and in many other VISTA manuals as well), when we talk about options for printing information, we are also talking about options for displaying the information on your screen, even if we don’t explicitly say so.

Printing a Default Report

In the previous chapter, we used the Inquire to File Entries option. That option asked you for a file, and then asked you to specify records from that file.

The Print File Entries option works a little differently. When you choose the file, Fileman assumes you want to print the whole thing. You can choose not to, of course, but that is the starting assumption. With that in mind, let's look at printing our imaginary COMPOSER file.

```
Select OPTION: PRINT FILE ENTRIES <Enter>

OUTPUT FROM WHAT FILE: COMPOSER <Enter>
Sort by: NAME// <Enter>
Start with NAME: FIRST// <Enter>
First print field: NAME <Enter>
Then print field: BIRTHDATE <Enter>
Then print field: BIRTHPLACE <Enter>
Then print field: <Enter>
Heading (S/C): COMPOSER LIST// <Enter>
DEVICE: <Enter>
```

Once we choose our file, Fileman asks how we want it sorted. We could sort by any field in the file, but for now, we're accepting the default and sorting by NAME. Next, Fileman asks us where we want to start. We could use this prompt to select only a certain range of composers, which we'll explore in greater detail a little later in the chapter. For now, we'll accept the default of FIRST, meaning we want to start with the first composer on the list.

Fileman then asks us to specify, one at a time, which fields we would like to see on our printed report. In our example, we picked three, but you could choose to show as many fields as you needed. When we're done choosing fields, we press the return key without entering any information.

Finally, Fileman asks us about the heading that should appear at the top of our report, and which device to send it to. We'll accept the default heading, and the default device, which is our computer screen.

Fileman's default format for printing files is a table, with the field names as column heads. Each page of the report has a header containing the file name, the date and time the report was printed, and the page number.

COMPOSER LIST	NOV 14, 2012	10:32	PAGE 1
NAME	BIRTHDATE		BIRTHPLACE
BACH, CARL PHILIPP EMANUEL	MARCH 8, 1714		WEIMAR
BACH, JOHANN CHRISTIAN	SEPTEMBER 5, 1735		LEIPZIG
BACH, JOHANN SEBASTIAN	MARCH 31, 1685		EISENACH
BARBER, SAMUEL	MARCH 9, 1910		WEST CHESTER
BEETHOVEN, LUDVIG VAN	DECEMBER 1770		BONN
BERNSTEIN, LEONARD	AUGUST 25, 1918		LAWRENCE

Our list of composers would go on longer, but this gives the general idea of what it would look like.

The Print File Entries option provides, broadly, two tools for creating customized reports: the “sort by” prompt and the “print field” prompt. Let’s look at both of these in a little more detail.

Custom Reports: Sorting

As we mentioned above, you can sort by any field you choose. You can also start your printout with any value of the field you choose. For example, we could sort our composers by birthdate, and limit our list to those born after 1900:

```
Sort by: NAME// BIRTHDATE <Enter>
Start with BIRTHDATE: FIRST// JANUARY 1, 1901 <Enter>
Go to BIRTHDATE: LAST// <Enter>
Within BIRTHDATE, Sort by: NAME <Enter>
First print field: NAME <Enter>
Then print field: BIRTHDATE <Enter>
Then print field: BIRTHPLACE <Enter>
Then print field: <Enter>
```

There are a couple of interesting things happening here. Notice that when we entered something other than FIRST as a response to “start with,” Fileman next asked us where we wanted to stop, with a default answer of LAST.

Also, notice that Fileman asked us how we wanted to sort “within

BIRTHDATE.” Essentially, this is Fileman’s way of asking, “if I find two composers with the same birthdate, what should I do?” That seems clear enough, but why is it asking us this when it didn’t ask before?

In our earlier sort, we accepted the default “sort by” field, which is NAME. As it turns out, we accepted more than that; we also agreed to let Fileman figure out for itself how to sort “within NAME.” When we chose a different field, Fileman also had to ask us what to sort by next.

One last thing to notice: it’s very unlikely that there’s actually a composer with a birthdate of January 1, 1901. That’s all right; that’s just the date where we want Fileman to start looking. It won’t get confused if there are no entries with that exact field value.

The resulting list would look like this:

COMPOSER NAME	NOV 14, 2012 BIRTHDATE	11:51	PAGE 1 BIRTHPLACE
-----	-----	-----	-----
KHACHATURIAN, ARAM	JUNE 6, 1903		TBLISI
SHOSTAKOVICH, DMITRI	SEPTEMBER 25, 1906		ST PETERSBURG
BARBER, SAMUEL	MARCH 9, 1910		WEST CHESTER
BRITTEN, BENJAMIN	NOVEMBER 22, 1913		LOWESTOFT
BERNSTEIN, LEONARD	AUGUST 25, 1918		LAWRENCE

...and so on.

The Nuts and Bolts of Sorting

We all have a vague idea of what it means to “sort” a list. For much of what you want to do, however, you probably want more than a vague idea. Here are some specifics about how sorting works in Fileman.

Fields are sorted from the lowest value to the highest value—what is often called “ascending order.” With some fields, such as numeric and date fields, this sort is fairly straightforward.

Set-of-code fields are sorted by their internally-stored code. It probably won't be immediately obvious.

Free text fields, as you might imagine, have a few things to keep in mind. First, the sort is case-sensitive, with capital letters sorting first. For example, BRAHMS would sort before Bach, because capital R sorts before lowercase a. It is also important to note that a free-text entry that contains only numeric characters (no alphabetic characters or punctuation) will be sorted as a number, and placed ahead of anything that includes letters or punctuation.

For any kind of field, Fileman will not include records with blank or null values in the sort. If, for example, we sorted our COMPOSER file by BIRTHPLACE, we would only see composers whose birthplace had some value filled in.

If you would like to include records with null values on your list, you can do that. At the "start with" prompt, enter an @ rather than FIRST. Records with null values in the sort field will appear at the top of the list.

Sorting with Subheads

Let's suppose we wanted to sort our list of composers by their era. There are only a few classical music eras, and it's of limited use to see a column that says, for example "BAROQUE" thirty times in a row. What we'd like is to sort the composers by era, but have them appear under era subheads rather than devote an entire column to them. Can we do this in Fileman?

We can! We can specify to sort by ERA, but then *not* choose ERA as one of the print fields, like this:

```
Sort by: NAME// ERA <Enter>
Start with ERA: FIRST// <Enter>
Within ERA, Sort by: NAME <Enter>
First print field: NAME <Enter>
Then print field: BIRTHDATE <Enter>
Then print field: <Enter>
```

Because ERA isn't one of our print fields, Fileman divides the resulting printout into sections, each with a subhead:

COMPOSER NAME	NOV 15, 2012 BIRTHDATE	09:36	PAGE 1

ERA: BAROQUE			
BACH, JOHANN SEBASTIAN	MARCH 31, 1685		
CORELLI, ARCANGELO	FEBRUARY 17, 1653		
HANDEL, GEORGE FRIDERIC	FEBRUARY 23, 1685		
PACHELBEL, JOHANN	AUGUST 1653		

...and so on. When Fileman finishes listing the baroque composers, it will print a new subhead for ERA: CLASSICAL, and list the classical composers.

Saving Sorts with Sort Templates

This sorting thing is kind of neat. Let's try another one.

```
Sort by: NAME// ERA <Enter>
Start with ERA: FIRST// <Enter>
Within ERA, Sort by: BIRTHDATE <Enter>
Start with BIRTHDATE: FIRST// <Enter>
Within BIRTHDATE, Sort by: NAME <Enter>
Start with NAME: FIRST// <Enter>
Store in 'SORT' Template:
```

Wait, what? Fileman is asking us a new question. Whenever you sort a file by three or more criteria, Fileman automatically asks if you want to save your sort. This is helpful if, for example, you want to perform a complex sort as part of a regularly-scheduled custom report. If you do not want to save the sort, press Return without entering information and Fileman will continue with the next prompt.

However, if we do want to save this sort, all we have to do is give it a name. We can pick just about any name we want, as long as it's less than 30 characters. Our template will be stored under our user ID, so we don't have to worry about choosing a name somebody else is already using.


```

Store in 'SORT' Template: PRACTICE SORT <Enter>
  Are you adding PRACTICE SORT as a new sort template? Y <Enter>
DESCRIPTION:
  No existing text.
  Edit? NO// <Enter>
First print field:

```

After we name our sort, Fileman asks us if we want to add a description. For now, we're just going to skip that, so we'll accept the default answer of NO. Fileman then continues with the dialog we have seen before, asking us which fields to print.

If you want to save a sort, you don't actually have to think of three fields to sort by. You can tell Fileman you want to save the sort before you start. You do this by entering a right square bracket at any "sort by" prompt.

```

Select OPTION: PRINT FILE ENTRIES <Enter>

OUTPUT FROM WHAT FILE: COMPOSER <Enter>
Sort by: NAME// ] <Enter>
Sort by: NAME//

```

After we type our square bracket, Fileman asks us again for our "sort by" criteria. At this point, it almost looks as if the square bracket had no effect! If we keep going, however, Fileman will prompt us for a Sort template name.

```

Sort by: NAME// ] <Enter>
Sort by: NAME// ERA <Enter>
Start with ERA: FIRST// <Enter>
Within ERA, sort by: NAME <Enter>
Start with NAME: FIRST// <Enter>
Store in 'SORT' Template:

```

In our examples of saving sorts, we have been accepting the default "start with" answer of FIRST. We should note, however, that if we chose something else, Fileman will have an additional question for us when we attempt to save our sort.

```

Sort by: NAME// ] <Enter>

```

```
Sort by: NAME// ERA <Enter>
Start with ERA: FIRST// @ <Enter>
Go to ERA: LAST// <Enter>
Within ERA, sort by: NAME <Enter>
Start with NAME: FIRST// <Enter>
Store in 'SORT' Template: ERA SORT <Enter>
  Are you adding ERA SORT as a new sort template? Y <Enter>
DESCRIPTION:
  No existing text.
  Edit? NO// <Enter>
Should template user be asked 'FROM-TO' range for ERA? NO//
```

Fileman gives us the option of saving the non-default from-to range we used. Or, we could choose to let the user specify each time. You might choose to save Sort templates in this way if you need to produce reports that are similar, but that cover different entries in the file.

Reusing Sorts

Now that we've saved a sort, how do we use it?

We call up our COMPOSER file again, and at the "sort by" prompt, we enter a left square bracket ([) plus the template name:

```
OUTPUT FROM WHAT FILE: COMPOSER <Enter>
Sort by: NAME// [PRACTICE SORT <Enter>
                      (Nov 16, 2012@09:17) User #XX File #YYY
Want to edit 'PRACTICE SORT' template? NO// <Enter>
First print field:
```

From there, we can designate the print fields and see our report.

Notice that when we responded to the "sort by" prompt, we entered the left square bracket before the template name, but not a right square bracket after it. Technically, there should be a right square bracket after the name of the template. Remember, however, that Fileman will accept a partial response, so we probably don't have to type that last bracket if we don't want to.

What if we forgot the name of our Sort template? We can ask Fileman. We

do this by entering a left bracket and a question mark ([?]) at the first “sort by” prompt:

```
OUTPUT FROM WHAT FILE: COMPOSER <Enter>
SORT BY: NAME// [? <Enter>
Answer with SORT TEMPLATE
Do you want the entire SORT TEMPLATE list? Y <Enter>
CHOOSE FROM:
  CAROL'S SORT
  KATHY'S SORT
  PRACTICE SORT
  RICK'S SORT
```

What’s going on here? Carol’s sort? Rick’s sort? Where did those come from?

A Sort template created by one user can be shared with other users. Entering [? at the prompt calls up a list of all the Sort templates available to you, whether you created them or whether somebody else did. After we choose PRACTICE SORT, Fileman prompts us to designate the print fields as before.

Hey, can’t we save the “print fields” part as well? We can, using another template. We’ll learn more about that in a moment.

Custom Reports: Choosing Print Fields

After you have completed the “select” process, Fileman asks you to list the fields you want displayed. We have already seen how to choose fields by typing their label. If you enter a single question mark (?) at the prompt, a list of all the fields in the file is displayed.

You can print all the fields in a file by entering ALL (uppercase) at the “First print field” prompt. Fileman next asks:

```
Do you mean ALL the fields in the file? NO//
```

Well, yeah! What a dumb question. What else could I mean?

Actually, there could be a field in the file that is called ALL, or that starts with ALL. Fileman is just making sure it understands what you mean.

Saving Print Criteria with Print Templates

We have already learned how to save sorts. Saving print criteria works much the same way. For example, if you choose five or more fields to print, Fileman automatically asks you for the name of a Print template you'd like to save. And, as with Sort templates, you can tell Fileman that you want to save your print criteria by entering a square bracket at any "print field" prompt. In the example below, assume we have already selected our sort criteria and have moved on to the "print field" prompts.

```

First print field: ]<Enter>
First print field: NAME<Enter>
Then print field: ERA<Enter>
Then print field: BIRTHDATE<Enter>
Then print field: <Enter>
Heading (S/C): COMPOSER LIST// <Enter>
Store print logic in template: PRACTICE PRINT<Enter>

```

Notice that our Print template stores not only which fields we print, but also what heading we want to use. In our example, we accepted the default heading. We'll learn how to customize headings later in this chapter.

Re-Using Print Criteria

Calling up and using a Print template works much the same way as using a Sort template. At the "first print field" prompt, enter a left square bracket and the name of the template you want to use. If you want to see a list of possible templates, enter a left square bracket followed by a question mark. Here is an example:

```

First print field: [ ?<Enter>
Answer with PRINT TEMPLATE
Do you want the entire PRINT TEMPLATE list? Y <Enter>
CHOOSE FROM:
  CAPTIONED
  ERA PRINT

```

PRACICE PRINT

As with the Sort templates, you see a list of all the Print templates you have access to.

The CAPTIONED Print Template

In addition to the Print templates you created, and the Print templates your co-workers have given you access to, there is a standard Print template that is available to everyone who uses Fileman. It is the CAPTIONED template, and it prints all fields, in an order established by Fileman.

Invoking the captioned template looks like this:

First print field: [CAPTIONED <Enter> Include COMPUTED fields (N/Y/R/B): NO//
--

We've seen the prompt about computed fields before, back in Chapter 5. We'll discuss what all of these options mean further in the manual; for now, you can just accept the default answer of NO.

Here's an interesting question. What's the difference between entering ALL at the "first print field" prompt, and entering [CAPTIONED at the "first print field" prompt?

The CAPTIONED Print template prints each field that has data, in order, just like our example in Chapter 5. When you answer with ALL, Fileman will still attempt to put all the fields on the same row, even if there are enough fields to make that impractical.

Custom Reports: Final Touches

After you establish how you're sorting your fields, and which fields you're printing, Fileman asks you about the heading, and then about the device you want to use. Each of these prompts gives you further opportunities to customize your report.

Report Headings

The heading in a report is what is printed at the top of each report page. Ordinarily, your heading consists of the report title, the date and time of the report, and a page number. It also includes column headers for the fields you selected as shown in the example below:

COMPOSER LIST		NOV 21,2012	11:10	PAGE 1
NAME	ERA			

The “headings” prompt gives you a few simple options for quick customization. For example, you might not know whether there will actually be any records that meet your selection criteria. If you enter an “S” at the “headings” prompt, you tell Fileman not to bother printing a heading at all if there aren’t any records found.

You can also enter a “C” at this prompt, and Fileman will print the sort criteria in the heading, like this:

COMPOSER LIST		NOV 21,2012	11:10	PAGE 1
Sort Criteria: BIRTHDATE from January 1, 1901 to LAST				
NAME	ERA			

You can use both of these together; entering “SC” tells Fileman to print the sort criteria, and to print no heading if there are no records to print.

You can also enter an “@” at this prompt to tell Fileman not to print any heading at all.

Custom Headings

You can completely customize the heading that appears at the top of your report. You do this with a Print template.

This seems a bit counterintuitive at first. Aren’t Print templates supposed

to hold which fields you print? The answer is “yes, but.” Yes, but they can also hold the information you would like to appear in your heading.

Your first step in creating a custom heading, then, is to get to where you can create and save a Print template. You can do this using either Inquire to File Entries or Print File Entries. Once you reach the “first print field” prompt, enter a right square bracket to tell Fileman you want to save a Print template. Then, enter the components of your heading in the “print field” prompts, like this:

```

First print field: ] <Enter>
First print field: "This is my heading " <Enter>
Then print field: "I made it myself " <Enter>
Then print field: "Hooray for me!" <Enter>
Then print field: <Enter>
Heading (S/C): COMPOSER LIST// <Enter>
Store print logic in template: MY HEADING <Enter>

```

Notice that the components of our heading are enclosed in quotation marks, so that Fileman knows we just want those words, and that we aren’t trying to refer to some file or field that Fileman needs to go look for. Also, notice that we included spaces, so that the parts of our heading would be spaced apart a little.

It would be nice if, at the “heading” prompt, you could somehow tell Fileman to use the Print template you’re working on right now. But that’s a bit too recursive for Fileman. Instead, we’ll accept the default heading and save the Print template. At the next prompt, we can enter a caret (^) to jump out, since we don’t actually want to print this right now.

To use this template for a heading, go back into Inquire or Print to get your report. Select your sort and print fields as normal. Then, at the “heading” prompt, enter the name of your template. Don’t forget the left square bracket!

```

Heading (S/C): COMPOSER LIST// [MY HEADING <Enter>

```

The resulting report would look something like this:

This is my heading	I made it myself	Hooray for me!
--------------------	------------------	----------------

After that heading, the data would begin immediately. Fileman would not print the column headers, or the line of dashes that we've seen in other heading examples.

Why is that? When you specify a heading, Fileman puts you in charge! You control everything. You can specify a multiple-line heading if you like, and your own column headers, and whatever else you want.

The downside is, nothing is done automatically. If you want column headers, you need to include them in your print template. If you want a line of dashes (or equal signs, or asterisks, or anything else), you'll need to specify it. You can control how your heading looks through the use of print qualifiers, which we will discuss in the next chapter.

Index Headings

In a dictionary, each page has a heading that includes the first and last word that can be found on the page. These are called index headings, and they are very useful when you're looking through a long list of data, such as dictionary entries.

You can create index headings in Fileman, which might come in handy if you have to print a long report that covers several pages. To create an index heading, create a Print template that includes the field you want to use for your index. For example:

```

First print field: ] <Enter>
First print field: "Composer: " <Enter>
Then print field: NAME <Enter>
Then print field: <Enter>
Heading (S/C): COMPOSER LIST// <Enter>
Store print logic in template: COMPOSER INDEX <Enter>

```

Notice that NAME is not in quotation marks, because we want Fileman to print the current value of the field at the top of the page. When we use this

Print template for a heading, a page somewhere in the middle of the report would look like this:

Composer: MENDELSSOHN, FELIX	
MENDELSSOHN, FELIX	ROMANTIC
MONTEVERDI, CLAUDIO	BAROQUE
MOZART, WOLFGANG AMADEUS	CLASSICAL
OFFENBACH, JACQUES	ROMANTIC

Custom Footers

You can use a Print template to put a custom footer on your report. To tell Fileman that your template should be used as a footer and not a header, precede it with a minus sign.

Heading (S/C): COMPOSER LIST// -[MY FOOTER <Enter>
--

When you use a custom footer like this, Fileman suppresses the heading, just as it does when you use a custom heading. If you want your report to include both a heading and a footer, you will need to specify both.

Heading (S/C): COMPOSER LIST// [MY HEADING]-[MY FOOTER <Enter>
--

When you specify both a heading and a footer, you do need to use that right square bracket to close off the name of your heading. Otherwise, Fileman will think you're trying to call up a template called "My Heading-My Footer," which you probably don't have.

Chapter 7: More Inquire and Print Options

Remember Inquire to File Entries? We discussed it briefly in Chapter 5, essentially showing how to get a quick report by accepting the defaults. You can, of course, do more with this option, by responding differently to the prompts.

If you remember, here's how we selected the file and records we wanted to see:

```
Select OPTION: INQ <Enter>

OUTPUT FROM WHAT FILE: COMPOSER// <Enter>
Select COMPOSER NAME: MOZART, WOLFGANG AMADEUS
Another one: HAYDN, JOSEPH
Another one: BACH <Enter>
  1 BACH, CARL PHILIPP EMANUEL
  2 BACH, JOHANN CHRISTIAN
  3 BACH, JOHANN SEBASTIAN
Choose 1-3: 3 BACH, JOHANN SEBASTIAN
Another one: <Enter>
Standard .Captioned Output? Yes//
```

We took the default for that last question in Chapter 5, but what happens if we say NO?

```
Standard Captioned Output? Yes// No <Enter>
First print field:
```

Hey, look! We know that prompt! We can enter the name of a field to print, and Fileman responds by asking us "then print field," just as it did in the Print File Entries option. Or, we can use a Print template; it will work here just as it works in Print File Entries. We can even use the [CAPTIONED template.

Why would we use the [CAPTIONED template instead of answering "yes" at the "standard captioned output" prompt? Answering "yes" at the prompt means not only accepting the template, but also accepting the default device (your computer screen). If you wanted to print out your

report on a printer, you would need to say “no” at the prompt, even if you wanted the standard captioning.

Computed Fields

Once you have chosen your print fields (whether in Inquire or Print), you are asked about computed fields. Until now, we have just accepted the default of “N.” Here are your other options, and what they mean.

You can answer “Y” to see all computed fields. This might include totals, subtotals, averages, and so on.

You can answer “R” to see the record number. Every record in your file has a record number, even though you normally do not see it. Fileman assigns these record numbers automatically every time a new record is entered. If you would like to see these record numbers, but not the other computed fields (averages and so on), this is the option to choose.

You can answer “B” to see both the computed fields and the record number assigned by Fileman.

What if you only want to see some of your computed fields, but not all of them? The place to handle that is not the “computed fields” prompt. You would want to select those fields at the “print fields” prompts, and create a Print template if you believe you’ll need to select the same fields again.

Choosing a Device

After choosing your print fields and answering the “computed fields” prompt, you are next asked for the output device. The default is your HOME device, which is usually your computer screen. You can, however, enter the name of a printer if you want to print out your report. You can also send your report to a spooler so that you can print more than one copy. If you type a question mark at the “device” prompt, you will see a list of

possible devices. If you are not sure which one you are supposed to use, contact your supervisor or IT department.

One device that may be available to you is the Browser. The Browser is a different way of viewing the data on your screen, which gives you more options and more flexibility than the simple “print to screen” we’ve seen so far. We discuss the Browser in greater detail in Chapter 9.

More Sorting Options

We’ve learned the basics of sorting, but Fileman has a lot more sorting tools at your disposal. There are control characters you can enter at the “sort by” prompt to further refine your sort so that it comes out the way you want it. A complete list of these *sort qualifiers* can be found in Appendix A. For now, here are a few of the most commonly used.

As we mentioned, Fileman sorts in ascending order. However, you can specify a reverse sort—that is, sort in descending order. Before you get too excited about this, we should mention that it only works for numeric and date/time fields. To sort our COMPOSER file by birthdate, in descending order, we would enter:

```
Sort by: NAME// -BIRTHDATE <Enter>
```

The minus sign in front of the name of the field tells Fileman we want a reverse sort.

We can also select entries by criteria without sorting from them. In one of our examples from Chapter 6, we selected composers born after 1900. In order to do this, though, we sorted them by birthdate, so they were printed in birthdate order. We didn’t actually have to do that part; we could have simply extracted composers born after 1900, without re-ordering them by birthdate. To do that, we type an apostrophe before the field name, like this:

```
Sort by: NAME// 'BIRTHDATE <Enter>
```

Another handy qualifier lets you suppress the subheads when you don't print the sort field. In our subhead example from Chapter 6, our subheads were reasonably helpful, dividing composers into eras. But what if we'd tried to sort by birthdate without printing the BIRTHDATE field? We'd end up with something like this:

COMPOSER	NOV 20, 2012	08:39	PAGE 1
NAME	BIRTHPLACE		

	BIRTHDATE: FEBRUARY 17, 1653		
CORELLI, ARCANGELO	FUSIGNANO		
	BIRTHDATE: MAY 2, 1660		
SCARLATTI, ALESSANDRO	PALERMO		
	BIRTHDATE: MARCH 4, 1678		
VIVALDI, ANTONIO	VENICE		

Yuck. Those subheads aren't helpful at all. Fortunately, we can tell Fileman not to print them, by putting an @ before our field name, like this:

```
Sort by: NAME// @BIRTHDATE <Enter>
```

Now, Fileman will not print subheads, even if we don't choose BIRTHDATE as a print field.

These three examples are among the more popular sort qualifiers, but they are not the only ones. Appendix A gives you a complete list.

More Printing Options

Just as there are special sort qualifier characters, there are similar print qualifiers. A complete list of print qualifiers can be found in Appendix A. For now, here are some of the most commonly used.

To format your report, you can specify where you would like a particular field to begin. For example, you could enter:

```
First print field: NAME<Enter>  
Then print field: ERA;C20<Enter>  
Then print field: BIRTHDATE;C-15<Enter>  
Then print field: <Enter>
```

These commands are asking Fileman to print the ERA field starting 20 columns from the left margin, and to print the BIRTHDATE field 15 columns from the right margin. The “;C” after the field name tells Fileman we want to designate a column. We follow that with a positive integer to count from the left, or a negative integer to count from the right.

In this example, what would Fileman do if the NAME field for a particular record was more than 20 characters long? It would print out the entire NAME, then print the ERA and BIRTHDATE for that composer on the next line down. That’s probably not what we want.

You can specify that Fileman should cut off, or truncate, a field value at a specific length. That would look like this:

```
First print field: NAME;L18<Enter>  
Then print field: ERA;C20<Enter>  
Then print field: BIRTHDATE;C-15<Enter>  
Then print field: <Enter>
```

The “;L” after NAME specifies that the NAME field should be left-justified. Of course, NAME is a free-text field, which is going to be left-justified by default. However, we can also enter the maximum number of characters we want to see. In this case, we’re asking Fileman to cut off the name after the first 18 characters, which should give us a nice two-column cushion before the ERA field begins.

You can also use this command to left-justify fields that are normally aligned differently, such as numeric fields. You must always enter a number after the “;L” to tell Fileman the size of the field you want to see.

If one of the fields you want has kind of an awkward name, you can specify a different field label on the printout. Here is how you would do that:

```

First print field: NAME<Enter>
Then print field: BIRTHDATE;"Date_of_Birth"<Enter>
Then print field: BIRTHPLACE;"Place of Birth"<Enter>
Then print field: <Enter>

```

The specified label, enclosed in quotation marks, is what Fileman will put in the header, instead of the field name. Notice how we used underscores in "Date_of_Birth;" this ensures that the label will be printed on one line and will not wrap.

Please see Appendix A for a complete list of print qualifiers.

Editing Templates

Each time you retrieve a template, whether for printing or sorting, Fileman asks if you want to edit the template. If you answer "yes," Fileman first asks about the name of the template:

```

Sort by: NAME// [PRACTICE SORT <Enter>
                (Nov 26, 2012@13:43) User #XX File #YYY
Want to edit 'PRACTICE SORT' template? YES// <Enter>
Name: PRACTICE SORT//

```

If you press Return here, you accept the default and keep the template name the way it is. You can also enter an @ at this prompt to delete your template altogether.

If you type in a different name, you will be changing the name of your template; the old name will no longer be used. For example, you could enter:

```

Want to edit 'PRACTICE SORT' template? YES// <Enter>
Name: PRACTICE SORT//DEMO SORT <Enter>

```

If you did this, then when you were finished editing, you would have a sort template named DEMO SORT, but you would no longer have one named PRACTICE SORT. If you asked Fileman to use your PRACTICE

SORT template, you would get an error because it would no longer exist under that name.

For this example, let's keep our template name the way it is.

```
Sort by: NAME// [PRACTICE SORT <Enter>
                (Nov 26, 2012@13:43) User #XX File #YYY
Want to edit 'PRACTICE SORT' template? YES// <Enter>
Name: PRACTICE SORT// <Enter>
Read access: <Enter>
Write access: <Enter>
```

What's that about? Read access? Write access?

You can use these prompts to allow some of your co-workers to access your templates. There are several options here, some of which depend on what kind of access you yourself have. For now, we'll just accept the defaults. More information on this feature can be found in the Fileman Advanced User Manual.

Next, Fileman brings up the fields you selected, one at a time. (Again, this applies to both Print and Sort templates.)

```
Sort by: ERA// <Enter>
Start with ERA: FIRST// <Enter>
  Within ERA, sort by: NAME//
```

You can change any of these parameters, and Fileman will save your changes when you are done.

With Print templates specifically, there is an additional feature you can use. You can insert a new field at any point in the sequence by typing a caret before the field name. Here is an example:

```
First print field: NAME// <Enter>
Then print field: ERA// <Enter>
Then print field: BIRTHDATE// ^BIRTHPLACE <Enter>
Then print field: BIRTHDATE// <Enter>
```


When we use the caret with BIRTHPLACE, Fileman inserts that field before the BIRTHDATE field, and keeps the BIRTHDATE field as the default for the next field to print. By contrast, if we didn't use the caret, we would see this:

```
First print field: NAME// <Enter>
Then print field: ERA// <Enter>
Then print field: BIRTHDATE// BIRTHPLACE <Enter>
Then print field:
```

We'd have to type BIRTHDATE in the last field. In this example, it wouldn't be a big deal. If, however, you were trying to insert a print field in, say, the second position when you had five more fields to go...re-typing all those fields would get a little tedious. Using the caret to insert a field can save you time.

Once you have finished updating your fields, Fileman gives you another prompt:

```
Store in SORT Template:
```

There is no default answer here, for reasons we'll explain in a moment. You can, however, use spacebar recall to bring up the name of your template, or the new name of your template, if you changed it at the "Name" prompt. Alternatively, you can type a different name for your template. Typing a new template name here works much like the "save as" option in many commercial software packages. You will have a new template with the new name, but your old template will still be there.

Creating a Temporary Edit

You can edit a Sort or Print template, and use the template one time, without saving the changes. You might want to do this if you have a one-time need to create or print a report that is similar to a template you have saved, but would require a few changes.

To do this, enter "yes" at the "want to edit?" prompt. At the "Name"

prompt, keep the template name the way it is. Then make whatever changes to the fields you need in order to create your one-time report.

After you finish with the fields, you will see the “store in SORT template” prompt. Press **Return** here without entering any information. This is how you tell Fileman, “actually, don’t save my changes; just use them this one time but keep my template the way it was.”

This feature is why there is no default at the “store in” prompt. If there were a default, you wouldn’t have the option of pressing **Return** to not save your changes; **Return** would instead accept the default answer.

Sharing Templates

Let’s suppose your department needed to produce a weekly report. Let’s also suppose that the VISTA packages your department is using don’t allow your users to generate that report.

So, you go backstage in Fileman, using your super-user access, and you create Sort and Print templates that will pull the information you need, format it the way you want, and print it out with custom headings and footers. Your report is a thing of beauty.

Of course, you’re the only one who can use the darned thing! What you want now is to make your templates available to other users. And not just super-users; you want the regular users in your department to be able to generate this report whenever they need to, using your templates. Is there a way for you to do this?

Not exactly. There isn’t a way *you* can do this; however, somebody with programmer access can do it pretty easily. Once you have your templates the way you want them, you can ask someone from your IT department to create a new option for your users, which uses your templates to generate the report that you need. Even if your IT department is pretty busy (and most of them are), this should be about a 15-minute job.

Creating custom reports is one of the ways you can help your team make the most of VISTA.

Chapter 8: Search

We have already learned how to sort entries, and select a certain range within that sort. There are situations, however, when sort-and-select isn't going to get you the information you need.

For example, suppose we want to find out how many composers in our COMPOSER file have "Johann" (or "Johan") as a first name. We can't find this information by sorting; the NAME field sorts alphabetically by last name. We need to find a way to search within the NAME fields for the particular string we're after.

How Searching Works

Now might be a good time to remind you that most packages have some kind of search option available to end users. It's possible you can find the information you're looking for without having to query Fileman directly.

If you do need to ask Fileman, you can use the Search File Entries option. Search File Entries has three basic steps: entering conditions, combining conditions, and formatting output. We'll go over each of these in detail.

Invoking this option looks like this:

```
Select OPTION: SEARCH FILE ENTRIES <Enter>
OUTPUT FROM WHAT FILE: COMPOSER <Enter>
  -A- Search for COMPOSER field:
```

Entering Conditions

Once you have selected the search option, and the file to be searched, your next step is to enter one or more search conditions to test each entry. These conditions are sometimes called *truth tests*, which is a fancy computer-language term for "yes or no question."

Notice that our prompt starts with the letter “A.” This is Fileman’s way of helping us keep track of which search condition we’re on. Our first condition is “A,” our second will be “B,” and so on.

The question we want to ask Fileman is: “does the NAME field contain the string ‘Johan’?” (The name “Johann” contains “Johan,” so no matter how they spell it, this query will find them.)

We would enter that search condition like this:

```
-A- Search for COMPOSER field: NAME <Enter>
-A- Condition: [ <Enter> Contains
-A- Contains NAME: JOHAN <Enter>
```

When we enter the left bracket, Fileman translates it to “contains,” which is the search condition we want. There are other symbols we can use for our search condition, as described in the table below.

Condition	Symbol	For Field Types
Null	NULL	All except word-processing.
Contains	[Numeric, free text, word-processing, MUMPS, set of codes, and computed fields.
Matches	MATCHES	Numeric, free text, MUMPS, computed, and date/time fields.
Less than	<	Numeric, computed, set of codes, free text, and date/time fields.
Equals	=	All data except word-processing.
Greater than	>	Numeric, computed, set of codes, free text, and date/time fields.

After we specify our “contains” search, Fileman prompts us for the “B” conditions, like this:

```
-A- Search for COMPOSER field: NAME <Enter>
-A- Condition: [ <Enter> Contains
```

```
-A- Contains NAME: JOHAN <Enter>
-B- Search for COMPOSER field: <Enter>
```

We don't have a "B" condition, so we press Return to tell Fileman that we're done entering conditions.

Combining Conditions

After the search conditions have been established, Fileman next asks how to combine them. There are several options, but let's start with our simple example.

```
-A- Search for COMPOSER field: NAME <Enter>
-A- Condition: [ <Enter> Contains
-A- Contains NAME: JOHAN <Enter>

-B- Search for COMPOSER field: <Enter>
  IF: A <Enter>
```

With the "if" prompt, Fileman is saying, "okay, I have the search conditions, now, how do I apply them?" Our response says, "if condition 'A' is true (that is, if the composer's name contains 'JOHAN'), then show me the record."

We could also enter 'A or -A, both of which mean "not A." That is, we could ask for a list of composers *not* named Johan or Johann.

After we enter A, Fileman does ask us for another possibility. Since our only other possibility, with just the one condition, is "not A," we'll go ahead and press Return to end this step.

```
IF: A <Enter>
OR: <Enter>
```

A Better Example

Don't you hate it when manuals give you the easiest possible example, then leave it up to you to figure out what to do when it gets complicated? Yeah, so do we. So let's come up with a more complicated set of criteria, and see how we would define them, and how we would combine them.

Let's say we're looking for a list of composers who are either named Johan or Johann, or are Romantic composers born after 1850, or were born in London.

(We have no idea *why* anyone would want this list, but in medicine you will do searches at least as complex as this, so it's a good example.)

The first thing to do, before we even touch the keyboard, is figure out the logic of what we're asking. We have three search conditions. The "Johan" one we already know how to do. We also have to look for composers born in London, which is a pretty straightforward search. We have to look for Romantic composers born after 1850....

You know what? We don't have three search conditions; we have four. Because "Romantic era" and "born after 1850" are two conditions, not one, even though we intend to combine them. This is why it's a good idea to work these things out ahead of time.

Now that we're (pretty) sure we know what we're asking, let's define our conditions.

```
-A- Search for COMPOSER field: NAME <Enter>
-A- Condition: [ <Enter> Contains
-A- Contains NAME: JOHAN <Enter>

-B- Search for COMPOSER field: BIRTHPLACE <Enter>
-B- Condition: = <Enter> Equals
-B- Equals BIRTHPLACE: LONDON <Enter>

-C- Search for COMPOSER field: ERA <Enter>
-C- Condition: = <Enter> Equals
```

```
-C- Equals ERA: ROMANTIC <Enter>

-D- Search for COMPOSER field: BIRTHDATE <Enter>
-D- Condition: > <Enter> Greater than
-D- Greater than BIRTHDATE: 12/31/1850

-E- Search for COMPOSER field: <Enter>

IF:
```

That was a lot of information, so let's highlight a few things. First, we chose the "equals" condition (=) for the birthplace. This means that Fileman will only show us records where the birthplace matches "London" exactly. If somebody made an entry and misspelled London, that record will not be selected.

If it was really important to get every record, we could have used the "contains" condition instead, and ask Fileman for records that contain the string "Lond" (for example). That might get some misspellings that the "equals" condition missed. However, it might also give us other cities with similar names, such as New London and Londonderry.

Which condition you choose for your search depends on many factors. How important is it that you get every single record that might fit the criteria? How many "extra" records will you get with a looser condition? How likely is it that you will miss records with the stricter condition?

These questions are related to a broader issue, which is database integrity. If you're having to go through a lot of extra steps to catch all the records you want, because information is misspelled, it may be time for a database cleanup. Some options for this are discussed in Chapter 10.

In this case, we decided that London is less likely to be misspelled than, say, Albuquerque, so we're taking a chance with the "equals" condition.

In our imaginary COMPOSER file, we're imagining that ERA is a set-of-codes field, rather than a free-text field. When searching a set-of-codes field, you must search on the actual value, not the code that represents it. In

our search, therefore, we used the whole word (ROMANTIC) rather than its code (R).

Now that we've established our conditions, let's tell Fileman how to combine them:

```
IF: A <Enter>
OR: B <Enter>
OR: C&D <Enter>
OR: <Enter>
```

What we're saying here is that we'd like to see the records where condition A is true, or condition B is true, or conditions C and D are *both* true.

When combining conditions, there are a few symbols you can use to tell Fileman what you want to see:

Condition	Symbol	Description	Example
AND	&	The conditions on both sides of the AND operator must be true. The "&" symbol can be omitted (i.e., AB is the same as A&B).	A&B
NOT	' or -	The condition following NOT must be false. If A is false, 'A evaluates as true.	'A
OR	Enter on new line.	Only one of the conditions combined with OR needs to be true. If A is true and B is false, A OR B evaluates as true.	IF: A <Enter> OR: B <Enter>

The difficult thing about creating and combining search conditions is not using the correct symbols. The difficult thing is training yourself to think with the formal logic of a computer. We wish we could help with that, but experience is really the only way to learn how to do this. If you want to be a whiz at finding the data you need, practice doing searches.

Search Templates

Just as you were able to save sort templates and print templates, you can also save your search specifications in a Search template. Fileman prompts you for the name of a template after you finish combining your search conditions:

```
Store results of search in template:
```

If you do not want to save your search, you can just press Return to continue to the next step. Otherwise, you can save your search by giving your template a name. Fileman asks if you want to save your new template, then prompts you for a description, in the same way we've seen with other kinds of templates:

```
Store results of search in template: PRACTICE SEARCH <Enter>
Are you adding 'PRACTICE SEARCH' as a new SORT TEMPLATE? No// Y
<Enter>
Description:
```

From here, you can enter a description and save your Sort template.

Whoa! Wait, hang on here. Sort template? We wanted to store a Search template!

Yes, we have uncovered a secret. To Fileman, a Search template is merely a specialized kind of Sort template. Every now and again, a VISTA expert will loftily inform you that "there's no such thing as a Search template." Technically, they're correct. However, you probably want your Search templates to do different kinds of things from your Sort templates, so it's perfectly okay to think of them as different kinds of templates. Even if not everybody agrees with you.

Formatting Output

We've specified our search conditions, and told Fileman how to combine them, and saved (or not saved) our search. What comes next?

```
IF: A <Enter>
OR: B <Enter>
OR: C&D <Enter>
OR: <Enter>

Store results of search in template: <Enter>

Sort by: NAME//
```

Well, this is a familiar sight. Once you've finished combining your search conditions, the rest of it is the same sort and print prompts we've been working with for the past few chapters.

An interesting aspect of this is that you can augment your search with the same sort-and-select options we've already discussed in Chapters 6 and 7. If you choose this option, Fileman does the sort-and-select *first*, then runs your search on the resulting selected list.

Print Number of Matches Found

Returning to our original question, the one that started the chapter, we wanted to know how many composers were named Johan or Johann. We didn't necessarily need to see a list of them; we just wanted to know how many there were.

We could, of course, just print them out and count them manually, but Fileman has another option for us. To print the number of matches found, without printing any of the matched entries, we can just press Return at the "first print field" prompt. That is, we don't specify any fields to print. Fileman responds with the number of records found, without actually showing them to us.

Using Your Templates

Once you have created a Search template, you have two options for how to use it. You can use it in the “Search File Entries” option like this:

```
Select OPTION: SEARCH FILE ENTRIES <Enter>
OUTPUT FROM WHAT FILE: COMPOSER <Enter>
-A- Search for COMPOSER field: [PRACTICE SEARCH] <Enter>
```

But remember what we said earlier. A Search template is really just a special kind of Sort template. Because of that, you can also use your Search template in the “Print File Entries” option, like this:

```
Select OPTION: PRINT FILE ENTRIES <Enter>
OUTPUT FROM WHAT FILE: COMPOSER <Enter>
Sort by: NAME// [PRACTICE SEARCH] <Enter>
```

The important thing to keep in mind with search templates is that you may see different results with the different options.

When you use a Search template in the “Search File Entries” option, you are instructing Fileman to *run your search query again*. If the file has been updated since the last time you used the search, and some of the new records meet your search criteria, you will see these new records in the results.

When you use a Search template in the “Print File Entries” option, you are instructing Fileman to *use the results obtained the last time the search was saved*. If the file has been updated since the last time you saved the search, you will not see any of the new records, even if they meet the search criteria.

A Search template can therefore be used to create and save a snapshot of data, which will remain the same no matter how many new records get added to a file.

Updating a Snapshot

There are times when creating a snapshot of key data is pretty darned handy. But what if we wanted to update our snapshot to include new data?

We can do this using the “Search File Entries” option. We call up the option, and use our PRACTICE SEARCH template, like this:

```
Select OPTION: SEARCH FILE ENTRIES <Enter>
OUTPUT FROM WHAT FILE: COMPOSER <Enter>
  -A- Search for COMPOSER field: [PRACTICE SEARCH] <Enter>
Store results of search in template:
```

After we specify our search template, Fileman asks us how to store the results. This prompt is the key to whether or not our snapshot gets updated. If we press Return here without entering any information, Fileman shows us the updated search results, but does *not* update the snapshot.

If we instead enter PRACTICE SEARCH, Fileman has a new prompt for us:

```
-A- Search for COMPOSER field: [PRACTICE SEARCH] <Enter>
Store results of search in template: PRACTICE SEARCH <Enter>
Data already stored there...OK to purge? NO//
```

If we answer “YES” at this prompt, Fileman will update our snapshot.

Searching Multiples

In the User section of this manual, we mentioned select prompts. Select prompts are often used with *multiples*, fields within a record that can store more than one value. For example, a patient record might have a DIAGNOSIS field, set up as a multiple so it could contain several different diagnoses. In our imaginary COMPOSER file, let’s imagine that each composer has a field called OPUS, set up as a multiple, which lists the composer’s works.

In a multiple, each entry or subfile can have its own subfields. For example, our OPUS field might have subfields for the opus name, composition date, and type of piece.

Searching on multiples is a special situation. If one of the fields in your search conditions is a multiple, Fileman prompts you for additional information, like this:

```
-A- Search for COMPOSER field: OPUS <Enter>

  -A- Search for COMPOSER OPUS sub-field: NAME <Enter>
  -A- Condition: [ <Enter> Contains
  -A- Contains NAME: SYMPHONY #9 <Enter>
```

In this example, we are searching for composers who have written a Ninth Symphony. Once we complete our search conditions, and combine them, Fileman has some additional questions for us:

```
IF: A <Enter>
OR: <Enter>

Do you want this search specification to be considered true for
condition -A-
1 When at least one of the OPUS multiples satisfies it?
2 When all of the OPUS multiples satisfy it?
Choose 1-2: 1//
```

Our choices, then, are to choose composers who have at least one work called Symphony #9, or composers whose works are all called Symphony #9. For this example, the choice is pretty obvious. For some of the medical data you may end up searching for, the choice may be less obvious. Remember what we said earlier: practice this stuff to get good at it.

Chapter 9: Viewing Reports on the Browser

If your site is using Kernel (if you are running VISTA, you are using Kernel), you may have the option of sending your reports (from the Inquire, Print, or Search options) to a device called the Browser. The Browser is an alternate method for viewing reports on your screen; it has more viewing options available than the basic “print to screen.”

You can find out whether the Browser is available to you by typing two question marks at the “device” prompt to get a list of available devices. If you do not have the Browser available, and would like to try it, ask your IT department about getting it enabled.

The Browser Screen

Let’s begin by taking a quick look at the Browser screen. As you can see, it has a lot in common with the word-processing Screen Editor.

Example		
COMPOSER	NOV 15, 2012 09:36	PAGE 1
NAME	BIRTHDATE	

ERA: BAROQUE		
BACH, JOHANN SEBASTIAN	MARCH 31, 1685	
CORELLI, ARCANGELO	FEBRUARY 17, 1653	
HANDEL, GEORGE FRIDERIC	FEBRUARY 23, 1685	
PACHELBEL, JOHANN	AUGUST 1653	
PURCELL, HENRY	SEPTEMBER 1659	
SCARLATTI, ALESSANDRO	MAY 2, 1660	
SCARLATTI, DOMENICO	OCTOBER 26, 1685	
TELEMANN, GEORG PHILIPP	MARCH 14, 1681	
VIVALDI, ANTONIO	MARCH 4, 1678	
ERA: CLASSICAL		
BACH, CARL PHILIPP EMANUAL	MARCH 8, 1714	
BACH, JOHANN CHRISTIAN	SEPTEMBER 5, 1735	
BOCCHERINI, LUIGI	FEBRUARY 19, 1743	
GLUCK, CHRISTOPH WILLIBALD	JULY 2, 1714	
HAYDN, JOSEPH	MARCH 31, 1732	
MOZART, WOLFGANG AMADEUS	JANUARY 27, 1756	
Col> 1 <PF1>H=help <PF1>E=exit Line>	22 of 300	Screen> 1 of 14

Browser Features

The Browser enables you to scroll back and forth through your data, a feature that is somewhat hit-and-miss in the traditional roll-and-scroll environment. Other features include copy-and-paste (into a different document), and the ability to search for a specific text string. The Browser's features are described in more detail below.

Navigating

As with Screenman applications and the Screen Editor, navigation in the Browser is handled from your keyboard. Here are the available commands:

To	Press
Scroll up or down one line at a time	<ArrowUp> and <ArrowDown>
Scroll right 22 columns	<ArrowRight>
Scroll to rightmost edge	<PF1><ArrowRight>
Scroll left 22 columns	<ArrowLeft>
Scroll to leftmost edge	<PF1><ArrowLeft>
Page Down	<PF1><ArrowDown>, <NextScreen>, or <PageDown>
Page Up	<PF1><ArrowUp>, <PrevScreen>, or <PageUp>
Go to Top	<PF1>T
Go to Bottom	<PF1>B
Go to specific line, screen, or column	<PF1>G At the prompt, enter a number, which you can precede with "S" (screen), "L" (line), or "C" (column). The cursor is re-positioned at the corresponding screen, line, or column. If you enter a number only, screen is assumed.

Exit	<PF1>E or <PF1>Q
------	------------------

Clipboard

The Browser uses Fileman's shared clipboard space, which means you can copy text from the Browser and paste it into any word-processing field using the Screen Editor. For example, you could generate a report, view it on the Browser, then copy some or all of your report and paste it into a Mailman message.

To use this feature, press <PF1>C. This should bring up the following prompt:

Copy Text Line(s) to Paste Buffer >

If you would like to copy all of the text in your report, you can enter an asterisk (*). Otherwise, you can enter the first and last line you would like to copy, separated by a colon, like this:

Copy Text Line(s) to Paste Buffer > 3:10

If you would like to append your text to what's already in the clipboard, rather than replacing what's in the clipboard, use the letter "A" with your command, like this:

Copy Text Line(s) to Paste Buffer > 3:10A
--

You can use the "A" with specific line numbers, or with the asterisk.

If at any time you would like to see what's already in the clipboard, press <PF1>V. The Browser will switch from your report to the clipboard, so you can see what's there. To return to your report, press <PF1>E.

You can paste the text from your clipboard into any word-processing field

using the Screen Editor. Place the cursor where you want the copied text to appear, and press <PF1>V.

Search

While in the Browser, you can search for a specific string. To use this feature, press <PF1>F (for Find). At the prompt, enter the string you would like to search for. You can specify the direction of the search by ending your string with <ArrowUp> or <ArrowDown>. If you just press the Return key after the string, the default search direction is down.

To find the same string again, you can press <PF1>N (for the Next occurrence of the string).

Online Help

As with most screen-based VISTA applications, the Browser includes online help. To access a Help summary, press <PF1>H. For more detailed help, press <PF1><PF1>H.

The Browser as an Option

If you like the Browser, you can also use it to view the contents of any word-processing field that you have access to. You can call up the Browser by choosing Other from Fileman's main menu, then selecting Browser from the Other submenu. Fileman then prompts you for the file, field, and entry to view.

Chapter 10: Editing Records

Records can be edited by ordinary users, using the options available to them in VISTA's packages. For most of the editing that you will need to do, these editing options should work just fine. However, you may need to use Fileman's more sophisticated editing tools to do database cleanup.

The .01 Field

You may have heard IT people, or other super-users, refer to "the .01 field." What does that mean? The .01 field is the single most important field in a file, as identified by the developer who created the file. For example, the .01 field of our imaginary COMPOSER file is NAME. The .01 field of a CLINIC file might be CLINIC ID NUMBER.

The .01 field plays a role in both adding and deleting records in a file. If you attempt to add a new record without filling in the .01 field, Fileman will not accept your record. To delete an entry, you can enter an at-sign (@) in the entry's .01 field.

Enter or Edit File Entries Option

You can edit data in a file directly using the "Enter or Edit File Entries" option. Keep in mind that you will only be allowed to edit a file if you have Write access to the file (if you need to review the different levels of access and what they mean, they are listed in Chapter 4). When you choose this option, Fileman asks for the name of the file you want to edit:

```
Select OPTION: Enter or Edit File Entries <Return>
Input to What File:
```

If you're not sure which files you're allowed to edit, you can enter two question marks at this prompt to get a list.

Let's assume we want to edit our trusty (and imaginary) COMPOSER file. Fileman next asks us which field(s) we'd like to edit:

```
Select OPTION: Enter or Edit File Entries <Return>
Input to what file: COMPOSER <Return>
Edit which field: ALL//
```

How you answer this prompt depends on what kind of database cleanup you're trying to do. If, for example, you're dealing with a problem where specific records are more or less completely screwed up, you would want to edit ALL fields for these records.

On the other hand, you may have a problem where one field consistently has wrong or incomplete information. In that situation, you would probably want to limit your editing to that one field.

For this first example, let's assume we want to edit all fields.

```
Select OPTION: Enter or Edit File Entries <Return>
Input to what file: COMPOSER <Return>
Edit which field: ALL// <Return>
Select COMPOSER NAME: RACHMANINOFF, SERGEI <Return>
```

Once the composer is selected, Fileman walks through all the fields, giving you the opportunity to edit them.

Wait, that's it? You can select a record by its .01 field, and then edit it? Can't you just do that as a regular user, in VISTA?

Yes, at its most basic level, "Enter or Edit File Entries" is a little underwhelming. So, let's not spend any more time at the basic level; let's learn about some of the options that make this a powerful tool.

Using ^LOOP

As long as you have to specify, one at a time, which records you want to edit, the "Enter or Edit File Entries" option isn't appreciably better than the

options available to regular VISTA users. Luckily, there's a way to tell Fileman that you want to look at a whole bunch of records. At the "select" prompt for choosing a record, enter the command **^LOOP**. This tells Fileman to go through (or loop through) a whole series of records, without you having to select each one.

```
Select OPTION: Enter or Edit File Entries <Return>
Input to what file: COMPOSER <Return>
Edit which field: ALL// <Return>
Select COMPOSER NAME: ^LOOP <Return>
  Edit entries by: NAME//
```

The "Edit entries by" prompt is very similar to the "Sort by" prompt we've seen before. We can use it to sort and select a group of records to be edited.

We've seen in at least one of our examples that not all the composers in our COMPOSER file have the ERA field filled in. Let's find those and fix them!

```
Select OPTION: Enter or Edit File Entries <Return>
Input to what file: COMPOSER <Return>
Edit which field: ALL// NAME <Return>
Then edit field: ERA <Return>
Then edit field: <Return>
Select COMPOSER NAME: ^LOOP <Return>
  Edit entries by: NAME// ERA <Return>
  Start with ERA: FIRST// @ <Return>
  Go to ERA: LAST// FIRST <Return>
```

There are a couple of things to note in this example. First, instead of opting to edit ALL fields, we're limiting our edit to the NAME and ERA fields. When we select a specific field (rather than ALL), Fileman asks us which field we'd like to edit next. If we wanted, we could choose several fields, which, depending on the situation, might be preferable to going through all the fields for each record we look at.

Why are we choosing to edit the NAME field? If we don't, we won't see the name of the composer we're editing! We'd just see a field labeled ERA, and we'd have to guess which composer it was.

It's been a while since the sorting chapter, so we'll remind you that the @ at

the “Start with” prompt tells Fileman to start with the null values, which are the ones we want to edit. The null values sort to the top of the list, so if we start with them, we want to end with FIRST; that is, with the first record that has a non-null ERA field.

How to Stuff a Field

This section is about “stuffing” field values, which honestly sounds a little rude. All it means, though, is that Fileman allows you to specify a default value for the fields you want to edit.

To illustrate, let’s revisit our ERA editing example. The Romantic era was pretty big for composers; it is the most likely response to the ERA prompt. To save a little time, we could specify ROMANTIC as our default answer during our edit, like this:

```
Edit which field: ALL// NAME <Return>
Then edit field: ERA//ROMANTIC <Return>
Then edit field: <Return>
```

After typing the name of the field, ERA, we enter two slashes, followed by the value we want to stuff into the field. Because we used two slashes, Fileman will consider ROMANTIC to be the default for any record where ERA is blank. Fileman would next show us the records we asked for, and we would have the opportunity with each record to either accept the default or enter something else.

Setting a default before editing can save time. But we would still have to go through each of the records we specified.

Let’s consider a different scenario. Earlier we hinted that “Albuquerque” is often misspelled. Let’s suppose that, at our facility, there were several patient records where the city was spelled “Albaquerque.” We want to find those, and stuff the correct spelling into each patient record. Can we do that, without having to manually go through each one?

We can! Using ^LOOP, we can specify the records we want to see. And by using three slashes instead of two, we can stuff the correct value into each record. This is what that would look like:

```
Select OPTION: Enter or Edit File Entries <Return>
Input to what file: DEMO PATIENT <Return>
Edit which field: ALL// CITY///ALBUQUERQUE <Return>
Then edit field: <Return>
Select DEMO PATIENT NAME: ^LOOP <Return>
  Edit entries by: NAME// CITY <Return>
  Start with CITY: FIRST// ALBAQ <Return>
  Go to CITY: LAST// ALBAQZ <Return>
```

As you can imagine, the ability to stuff values into fields is incredibly powerful. And it has the potential to create some truly spectacular errors, if it is not used carefully. For example, when we stuffed “Albuquerque” into the city field, we needed to be careful about which records we selected. If we had begun our edit with ALBA rather than ALBAQ, we could have moved hundreds of innocent patients from Albany to Albuquerque in an instant!

We can also use three slashes to delete all field values for the records we select. To do this, we follow the three slashes with an @, like this:

```
Then edit field: ERA///@ <Return>
  WARNING: THIS MEANS AUTOMATIC DELETION!
```

If we use the @ to stuff a null value, Fileman warns us that this will delete information that may already be entered in the field.

Using Templates

We mentioned earlier that the “Edit entries by” prompt you receive when you use ^LOOP is very similar to the “Sort by” prompts we’ve seen before. They are so similar, in fact, that you can actually use a Sort template with ^LOOP to choose the fields you want to edit, and the records within those fields. This is a handy option if you happen to have a Sort template that

specifies the fields and records you want to edit. However, there is also a template specifically designed for use while editing.

Input Templates

By now we've learned how to create Sort templates, Print templates, and Search templates (which do exist for our purposes). Creating an Input template is very similar to what we've seen already.

If you select at least five different fields to edit, Fileman automatically prompts you for a template name. You can also specify the creation of a template by answering the "Edit which field" prompt with a right bracket.

Once you have your template saved, you can use it by entering its name, enclosed in brackets, at the "Edit which field" prompt. As with other templates we've seen so far, Fileman asks if you would like to edit the template before you use it. Editing an Input template is similar to editing a Sort or Print template. Chapter 7 has specifics of how this process works.

If you edit your Input template, you can insert a new field using the caret or up-arrow character ("^"), just as you can with a Print template.

In Chapter 7, we discussed how your IT department can use your Sort and Print templates to create new report options for your users. Similarly, your IT department can create new options based on your Input templates, so that your users have new options for how they enter and edit records.

Edit Qualifiers

In Chapter 7, we discussed sort and print qualifiers, which you can use to refine and format your reports. You can also use qualifiers when editing. Most of these qualifiers are intended to be used in Input templates that you share with users; they help define what your users see and how they enter information.

Qualifier	Action
field;"xxx"	Replace the field's label with a literal string during an editing session.
field;T	Replace a field's label with its title during an editing session.
field;REQ	Require a response to a field that is usually not required.
field;DUP	Save responses for later use with spacebar recall.

You can combine specifiers by separating them with semicolons, like this:

```
Then edit field: DATE OF BIRTH;T;REQ <Return>
```

Creating Special Prompts

Normally, Fileman refers to fields by their *label*. A label is a specific attribute belonging to a field, which is set by the programmer when the field is first created. The field label is what you usually see when sorting, editing, or printing files. For example, our imaginary COMPOSER file has fields labeled NAME, ERA, BIRTHDATE, and so on.

Two of the edit qualifiers allow you to show the user a field to edit, but use something other than the field label to tell the user which field it is. For example, we might have the user edit the BIRTHDATE field, but specify different text for them to see, like this:

```
Edit which field: BIRTHDATE;"DOB" <Return>
```

Then the user would see this:

```
DOB:
```

Why would you do this? Maybe you have reason to think that your users (or some of your users) will be editing on a small and/or crowded screen, and you want to minimize the amount of space your prompt uses.

In addition to the label, fields can have titles. Often, the title is another name for the same information. You can choose to show the user the field's title instead of its label by using "T" after the field, like this:

```
Edit which field: BIRTHDATE;T <Enter>
```

If you used that command for a template, the user would see the title of the field, instead of the label, when they were editing or entering records.

How do you know what a field's title is? There's a way you can view this, and other attributes belonging to a field, but it's a little more advanced, and often requires a higher level of file permission. File attributes are covered in more detail in the *Fileman 22.2 Advanced User Manual*.

For now, you can see a field's title by calling up the "Enter or Edit File Entries" option, and using the "T" qualifier when editing the field. You may just see the label, which means that this particular field does not have a title.

Requiring Input

As we discussed in Chapter 3, some fields are required by Fileman. This kind of requirement is set by programmers.

However, you, as a super-user, can make certain fields required within your template. This does *not* change their status in Fileman; Fileman will still regard them as non-required fields. You can, however, ensure that your users won't be able to leave these fields blank. The specification looks like this:

```
Edit which field: NAME;REQ <Enter>
```

The REQ qualifier is a good one to use if a specific piece of information is badly needed in your department, but perhaps not in the facility as a whole. You can ensure that your users have to include it, without interfering with the practices of other departments.

Duplicating Input Values (Spacebar Recall)

Sometimes many entries will need the same data value input for a particular field. If you follow a field label with ;DUP, then your users can use spacebar recall to repeat their previous answer (spacebar recall is described in Chapter 3). For example, we could say:

Edit which field: ERA;DUP <Enter>
--

The users would then be able to use spacebar recall to repeat their earlier response. Since there are a limited number of answers to this field, spacebar recall could save the users a bit of time.

By now, we've seen a few ways you can manage user input in your templates. The following chart summarizes some things you might want to do for your users, and what commands you would use to make it happen.

I want to.....	Command	
Set up a default that users see when the field is blank.	Two-slash stuff: <table border="1"><tr><td>ERA//ROMANTIC</td></tr></table>	ERA//ROMANTIC
ERA//ROMANTIC		
Allow spacebar recall.	Duplicate qualifier: <table border="1"><tr><td>ERA;DUP</td></tr></table>	ERA;DUP
ERA;DUP		
Place a value in the field without user action.	Three-slash stuff: <table border="1"><tr><td>DEPARTMENT///BILLING</td></tr></table>	DEPARTMENT///BILLING
DEPARTMENT///BILLING		

I want to.....	Command
Require that a value be entered.	Required qualifier: <input data-bbox="875 422 1419 464" type="text" value="ERA;REQ"/>
Delete all values from the field without user action.	Three-slash stuff of @: <input data-bbox="875 600 1419 642" type="text" value="ERA///@"/>
Require that a value be entered <i>and</i> allow spacebar recall.	Edit qualifiers can be combined! <input data-bbox="875 779 1419 821" type="text" value="ERA;REQ;DUP"/>

Your Next Steps

As its title suggests, this manual is meant to get you started with the kinds of Fileman tasks you are most likely to perform as a super-user. Once you have finished with this manual, you have a few different options of where to go next.

If you are interested in more information about the packages your department users, those packages should have their own documentation, including information for super-users.

If you are interested in learning more about what super-users can do in Fileman, you will find it in the Fileman *Advanced Users' Manual*.

If the kinds of things you want to do seem to involve programming, you could investigate the Fileman *Programmers' Getting Started Manual*. You will probably need additional permissions if you are going to take on programming tasks.

Appendix A: Sort and Print Qualifiers

Sort Qualifier Reference

The following table illustrates the sort qualifiers you can use at the “SORT BY:” prompts:

Qualifier	Action	Discussion
!field	Number entries by sort value	Entries having the same value for the SORT BY field are numbered sequentially starting at one.
#field	Page break when sort value changes	Starts a new page every time the value of the sorted field changes.
'field	Specify a range for output without sorting	Lets you select entries based on this field in the “START WITH . . . GO TO . . .” dialogue, but not sort based on the field.
+field	Subtotal within a SORT BY field	<p>You can sub-subtotal by adding a prefix of + to two fields in the “SORT BY:” dialogue. For example:</p> <div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> <p>SORT BY: +REGION WITHIN REGION, SORT BY: +STORE</p> </div> <p>This produces subtotals for each STORE within the subtotals for each REGION.</p>
-field	Reverse sort order	Only effective for NUMERIC and DATE/TIME valued fields.
@field	Suppress the printing of subheader	Normally, a subheader with the value of the SORT BY field name is printed if the field is not also specified as a print field.
field;Cn	Set column of	The number (n) indicates the column to

	subheader	begin the subheader.
field;Ln	Use less than the entire sort field for sorting	<p>The ;L suffix lets you specify that only the first “n” characters of the field be used to sort. Thus, if you specify:</p> <pre style="border: 1px solid black; padding: 5px;">SORT BY: NAME;L2</pre> <p>Only the first two letters of the name field will be used for sorting. The order of “Maris,Roger” and “Mantle,Mickey,” for example, will be unpredictable in your output.</p>
field;Pa-b	Store sort range for display	<p>The ;P suffix saves the “START WITH” and “GO TO” sort range for one level of sort fields, for later reference at a “PRINT FIELD” prompt. You can substitute any string (up to 60 characters in length) for “a” and “b” in “;Pa-b.” Whatever strings you use as “a” and “b” (e.g., FROM and TO) become subscripts of the PARAM() array:</p> <pre style="border: 1px solid black; padding: 5px;">SORT BY: NAME;PFROM-TO START WITH NAME: FIRST// C GO TO NAME: LAST// Dzzzzz</pre> <p>Then at a “PRINT FIELD” prompt, you can reference the saved values by using PARAM(“a-string”) and PARAM(“b-string”). Thus, in this case:</p> <pre style="border: 1px solid black; padding: 5px;">THEN PRINT FIELD: PARAM("FROM") THEN PRINT FIELD: PARAM("TO")</pre> <p>The sort-from and sort-to values of “C” and “Dzzzzz” in the report would then print in the appropriate location. The typical reason</p>

		<p>to save sort values, and then retrieve them from the PARAM array in a print field, is for custom report headers (which are created as PRINT templates). It is one method to print sort criteria in the report header.</p> <p>If you need to sort on two or more fields, you need to use a different string for "a" and "b" at each level, or else the values will overwrite each other.</p> <p>No value is saved in the PARAM array if the user accepts "FIRST" or "LAST" as the "START WITH" or "GO TO" sort values; the PARAM array will be null.</p>
field;Sn	Skip lines when sort value changes	<p>The ;S suffix skips one line between sort field sort value values, if no number is added. If you specify changes a number (n), n-lines are skipped. For example, here we sort by patients, but we ask that each time the patient value changes, we add 3 lines.</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <pre>Output from what File: VISIT// Sort by: VISIT/ADMIT DATE&TIME// PATIENT NAME;S3 Start with PATIENT NAME: FIRST// Within PATIENT NAME, Sort by:</pre> </div>
field;TXT	Sort numbers as text	<p>The ;TXT sort qualifier forces digits to be sorted as alphanumeric strings, not as numbers (22 sorts before 3). Use this to correctly sort mixed canonic and non-canonic numbers.</p>
field;"xxx"	Replace caption in subheader and sort	<p>The default subheader is the field label or expression entered at the "SORT BY:" prompt. The same string is used within the</p>

	dialogue	sort dialogue. You can substitute a string of your choosing by adding a semicolon (;) and text enclosed in quotation marks. For example: SORT BY: DATE OF BIRTH;"Birthdate: " To suppress the caption in the subheader, enter two double quotes after the semicolon: SORT BY: DATE OF BIRTH;" The specified caption is also used in the subsequent sort dialogue. This is useful in simplifying the sort dialogue when a field label is confusing, is concatenated, or is an extended pointer. For example: SORT BY: STORE:REGION;"Location:" START WITH Location: FIRST / / <Enter> WITHIN Location, SORT BY: <Enter> PRINT FIELD: STORE THEN PRINT FIELD: "Location:" is substituted for "STORE:REGION."
--	----------	--

Print Qualifier Reference

The following table illustrates the print qualifiers you can use at the "PRINT FIELD:" prompts:

Qualifier	Action	Discussion
!field	Count	The number of entries with non-null values in this field are counted.
#field	All statistics	Use with NUMERIC fields only. Six lines of statistical output are generated for the selected field: Total, count, mean, maximum, minimum, and standard deviation.
&field	Total	Use for NUMERIC fields only. The values of all occurrences of this field are summed.
+field	Total, count and mean	Use for NUMERIC fields only. The sum, count of non-null values, and the arithmetic

		mean of the non-null values are printed.
field;Cn	Set start column position	You can specify the starting column of a field's display. If the number (n) is a positive number, the field prints in that column counting from the left margin. If it is negative, the field starts in that column as subtracted from the right margin. If there is not enough room on a line for the field to display properly, it is moved to the line below. Adding ;C1 always starts a field's display at the beginning of a line.
field;Dn	Round decimal fractions	Specify the number of decimal digits of accuracy with a number, "n." If n=0, the number is rounded to a whole number. The number (n) must be > 0 or = 0.
field;Ln	Left justify and truncate	To left justify within a field width of "n" characters, follow the ;L with the number of column positions you want the field to occupy. If necessary, the data is truncated to fit into this length. Since non-numeric fields are left justified by default, the only effect of ;L on those fields is truncation. This qualifier does not affect DATE/TIME-valued fields.
field;N	Suppress consecutive duplicate values	If you do not want the same value for a field printed on consecutive rows of a report, add ;N to the field specification.
field;Rn	Right justify text	To right justify within a field of "n" characters, follow the ;R with that number. Right-justified data is not truncated; if "n" is not large enough, data spills into the next field's area causing confusing output. NUMERIC DATA TYPE fields are right justified by default. However, computed

		expressions with numeric results are not right justified automatically.
field;Sn	Skip lines between fields	<p>You can indicate a number of lines to skip before printing a field by using ;Sn. Without a number, ;S skips a single line. If your field has a null value, "n" lines are skipped anyway. You cannot use ;S with a WORD-PROCESSING-type field. To skip one line before a WORD-PROCESSING field, precede the field with an additional print specification like this:</p> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 5px auto;"> <p>THEN PRINT FIELD: " " ;S</p> </div>
field;T	Use field TITLE as header	If you want to use the field's TITLE, as defined in the data dictionary, instead of its LABEL in the header, add ;T.
field;Wn	Wrap text	The ;W suffix splits a field that is too long for its field column width at word boundaries (spaces) and prints it out fully on two or more rows. You can optionally follow the ;W with the number of column positions to be occupied by each line of output. If the field's data dictionary definition says to always print in word-wrap mode, WORD-PROCESSING DATA TYPEs are automatically wrapped. The ;W suffix overrides the ;R or ;L suffixes; do not use them together.
field;X	Suppress header and inter-column spaces	Follow the field specification with ;X to suppress the two spaces normally inserted before a field and the column header for the field. The result is concatenation of the field with the field that came before it. The ;X suffix is often used to add a literal caption to a field.

<p>field;"xxx"</p>	<p>Customize header</p>	<p>The default header for columns of a report is the field label (for fields) or a computed expression (for “on the fly” computed expressions). To change the header, follow the field label with a semicolon (;) and the column header you want. Enclose the header in quotes. When possible, spaces wrap the substitute header within a print width obtained from the field's data definition. For example:</p> <div data-bbox="678 747 1412 814" style="border: 1px solid black; padding: 5px;"> <pre>THEN PRINT FIELD: DOB;"Date of Birth" THEN PRINT FIELD: DOB;"Date_of_Birth"</pre> </div> <p>The first specification prints the header on two lines; the second prints it on one line (with underscores). To suppress the header, add ;"" to the field.</p>
<p>field;Yn</p>	<p>Set start row position</p>	<p>You can start printing a field on any line on the page. If “n” is positive, the field is printed on that line as counted from the top of the page. If it is negative, it is printed on that line as counted from the bottom. When specifying ;Yn, you must account for any page header that is displayed.</p>
<p>"xxx"</p>	<p>Insert a literal</p>	<p>At the “PRINT FIELD:” prompt enclose a free text literal in quotes. The literal is inserted into the body of the report. For example, the following specifications begin each entry's display with a caption and NAME:</p> <div data-bbox="678 1650 1412 1717" style="border: 1px solid black; padding: 5px;"> <pre>FIRST PRINT FIELD: "Patient's name:" THEN PRINT FIELD: NAME;" "</pre> </div>

		<p>OR</p> <table border="1"><tr><td><pre>FIRST PRINT FIELD: "Patient's name:" THEN PRINT FIELD: NAME;X</pre></td></tr></table> <p>The header for the NAME field is suppressed in both examples.</p>	<pre>FIRST PRINT FIELD: "Patient's name:" THEN PRINT FIELD: NAME;X</pre>
<pre>FIRST PRINT FIELD: "Patient's name:" THEN PRINT FIELD: NAME;X</pre>			

Glossary

.01 Field	The single most important field in a file, as defined by the file's creator.
Alternate Editor	One of the text editors available for use from Fileman. Editors available vary from site to site. They are entries in the ALTERNATE EDITOR file (#1.2).
Browser	A Screen Mode tool that allows you to view Fileman output. If enabled, the Browser can be selected at the device prompt or as a Fileman option.
Caption	In Screenman, a label displayed on the screen. Captions often identify fields that are to be edited.
Command Area	In Screenman, the bottom portion of the screen used to display help information and to accept user commands.
Data Dictionary	A record of a file's structure, its elements (fields and their attributes), and relationships to other files. Often abbreviated as DD.
Data Type	The kind of data stored in a field. Numeric, computed, and word-processing are examples of Fileman data types.
Database	An organized collection of data spanning many files. Often, all the files on a system constitute that system's database.
Default	A computer-provided response to a question or

prompt. In Fileman, the default is often the most common answer or the most recent answer.

Device Prompt	A prompt that allows you to specify where your output should be “printed.” Devices include your computer screen, the spooler, a network printer, and the Browser (if enabled).
Edit Window	In Screenman, the area in which you enter or edit data. It is highlighted with either reverse video or an underline. In Screen Editor, the area in which you enter and edit text; the area between the status bar and the ruler.
Entry	A record in a file. “Entry” and “record” are used interchangeably.
Entry Number	See Internal Entry Number.
Field	In a record, a specified area used to hold values. The specifications of each Fileman field are documented in the file’s data dictionary.
Field Number	The unique number used to identify a field in a file.
File	A set of related records (or entries) treated as a unit.
Form	In Screenman, a group of one or more pages that comprise a complete transaction.
Free Text	A data type that can contain any printable characters.
Identifier	A field used to identify a record, sometimes in conjunction with other identifiers. Some identifiers are required, meaning that the record will not be saved if the field is a null value.

Input Template	A template of fields to be edited, used in conjunction with the Enter or Edit File Entries option.
Internal Entry Number	The number used to identify an entry within a file. Every record has a unique internal entry number. Often abbreviated as IEN.
Kernel	A VISTA software package that functions as an intermediary between the host operating system and VISTA application packages. Kernel includes installation, menu, security, and device services.
Key	One or more fields that uniquely identify a record.
Label	An attribute of a field. The name you usually see when encountering the field (for example, NAME).
LAYGO	A user's authorization to create a new entry selecting a record from a file. An acronym for Learn As You Go.
Line Editor	The Fileman editor that lets you input and change text on a line-by-line basis. The Line Editor works in scrolling mode. See Screen Editor.
Mailman	An electronic mail system (e-mail) that allows you to send messages to and receive them from other users via the computer. It is part of VISTA.
Menu	A list that includes the names of options from which you can select an activity.
Multiple	A Fileman data type that allows more than one value for a single entry. See Subfile.
MUMPS	The computer language used by Fileman and throughout VISTA. The acronym MUMPS stands for

	Massachusetts General Hospital Utility Multi-Programming System. MUMPS is sometimes shortened to M.
Non-null	A value other than null. A space and zero are non-null values.
Null	Empty. A field or variable that has no value associated with it is null.
Option	A computing activity that you can select, usually a choice from a menu.
Pointer to a File	A field data type that “points” to information that resides in another file.
Pop-up Page	In Screenman, a page that overlays the regular Screenman screen in order to present the contents of a selected multiple.
Preferred Editor	The editor always entered when you access a word-processing field; your default editor. Kernel must be present to establish a Preferred Editor.
Print Template	The stored specifications of a printed report, including fields to be printed and formatting instructions.
Programmer Access	The ability to use Fileman features that are reserved for application developers. You do not need programmer access to use any of the features described in this manual.
Record	A set of data pertaining to a single entity in a file; an entry in a file.
Record Number	See Internal Entry Number.

Required Field	A field that cannot be left null for an entry.
Screen Editor	Fileman's Screen-oriented text editor. It can be used to enter data into any word-processing field using full-screen editing.
Screen Mode	A computer interface in which you see many lines of data at a time and in which you can move your cursor around the display screen using screen navigation commands. Compare to Scrolling Mode.
Screenman	The set of routines that supports Screen Mode.
Scrolling Mode	The presentation of the interactive dialogue one line at a time, often called "roll-and-scroll." Compare to Screen Mode.
Search Template	The saved results of a search operation. Usually, the actual entries found are stored in addition to the criteria used to select those entries.
Set of Codes	A field type where a short character string is defined to represent a longer value.
Sort Template	The stored record of sort specifications. It contains sorting order as well as restrictions on the selection of entries. Used to prepare entries for printing.
Subentry	A record in a multiple.
Subfield	A field in a multiple.
Subfile	A file within a file, having its own fields. Sometimes called a multiple field, or just a multiple.
Title	An attribute of a field, often an alternate name for the

	field. Although all fields can have titles, not all of them do. See Label.
Truth Test	An evaluation of an expression yielding a <i>true</i> or <i>false</i> result. Used in the Search File Entries option.
Up-arrow (caret)	The ^ character (caret); used in Fileman for exiting an option or canceling a response. Also used in combination with a field name or prompt to jump to the specified field or prompt.

Index

cancel.....	20	combining.....	58, 61
captioned output.....	29, 46	searching.....	
computed fields.....	9, 29, 47	number of matches.....	63
copy and paste (from Browser).....	69	select prompts.....	7, 16, 22
custom footers.....	45	set-of-codes fields.....	11, 60
custom headings.....	42	sort qualifiers.....	48, 81
date fields.....	9	sorting.....	33
default answer.....	7, 74	selecting.....	33, 48
deleting.....	21	sort order.....	34, 48
edit qualifiers.....	76	spacebar recall.....	22
file access.....	26	standard captioned output.....	29
free text fields.....	10, 35	subheads.....	35
graphical user interface (GUI).....	5p.	suppressing.....	49
identifiers.....	21	super-user.....	1, 25
index headings.....	44	templates.....	
jumping.....	19	captioned.....	41
jumping out.....	20	editing.....	51
key fields.....	21	input.....	76
list mode.....	5	print.....	40, 52
multi-part response.....	16	search.....	62
multiples.....	65	Search.....	64
numeric fields.....	11	sharing.....	54
partial response.....	16, 38	sort.....	36
pointer fields.....	11	truth test.....	56
print qualifiers.....	44, 49, 84	variable pointer fields.....	11
print to screen.....	31	word-processing fields.....	11
replace...with.....	17, 19	word-processing tools.....	
using ellipses.....	18	alternate editor.....	15
required fields.....	21	line editor.....	14
required identifiers.....	21	screen editor.....	12
search condition.....	57, 59	<PF1> key.....	13