



ScreenMan Tutorial For Developers

June 1996

Department of Veterans Affairs
VISTA Software Development
Infrastructure Product Line

Preface

This tutorial is intended to help **VISTA** developers create ScreenMan forms using VA FileMan's Form Editor. The Form Editor provides a major improvement in the speed and ease of creating ScreenMan forms. The tutorial emphasizes the use of the Form Editor in creating forms.

Acknowledgements

Thanks are due for invaluable comments, suggestions, and reviewing to:

- Fil Beza (IRM Chief, VA Northern California System of Clinics, now located at the San Francisco CIO Field Office)
- David Bolduc (Computer Specialist, AIM Center, St. Louis VA Medical Center)
- David Laliberte (Computer Specialist, IRM, White City Domiciliary)

Thanks also to David Laliberte for the screen capture of one of his ScreenMan forms, from his ASI (Addiction Severity Index) package.

Table of Contents

Introduction	1
Intended Audience.....	1
Objectives.....	1
Files Used	2
Advance Preparation.....	2
1. Starting Out with Forms.....	5
1-1. Create a Form.....	7
1-2. Add a DISPLAY Block to the Form.....	11
1-3. Add a Title (Caption-Only Field).....	15
1-4. Re-position Fields on a Block.....	17
1-5. Add Data Dictionary Fields	19
1-6. Running the Form.....	23
1-7. Add an Edit Block to the Form.....	25
1-8. Add Data Dictionary Fields to the EDIT Block	27
1-9. Use PRINT^DDS to List Form Characteristics.....	31
2. Pages, Multiples, and Computed Fields	35
2-1. Add a Second Page	37
2-2. Add a Multiple Field (in a Repeating Block).....	43
2-3. Add a Multiple Field (in a Pop-Up Page)	49
2-4. Reposition and Resize the Pop-Up Page	53
2-5. Add a ScreenMan Computed Field.....	55
3. Advanced Features for Your Forms	59
3-1. Edit Fields in Other Files	61
3-2. Add Branching Logic for Fields	65
3-3. Use Form-Only Fields for User Input	67
3-4. Turn Your ScreenMan Form into an Option.....	69
3.5. Where to Go From Here	73
3-6. Delete Your Form.....	75
Appendix A: M Code and Variables in ScreenMan Forms	77
Appendix B: Tutorial Keystroke Summary	81
Index	83

Table of Contents

Introduction

This tutorial is intended to help **VISTA** developers create ScreenMan forms using VA FileMan's Form Editor. The Form Editor provides a major improvement in the speed and ease of creating ScreenMan forms. The tutorial emphasizes the use of the Form Editor in creating forms.

The tutorial may take 10 to 20 hours to complete, depending on your familiarity with VA FileMan.

NOTE: There is an HTML version of this tutorial posted on the VA FileMan Home Page at:

<http://www.vista.med.va.gov/softserv/infrastr.uct/fileman/index.html>

Some of the lessons in HTML version of this tutorial come with companion movies, in Lotus ScreenCam format. The intent of including these movies is to provide a visual demonstration of the techniques discussed in each lesson.

Intended Audience

The intended audience for this tutorial is IRM developers at both Medical Centers and Field Offices (formerly known as Information Systems Centers). Users of this tutorial should be familiar with M programming, VA FileMan's data structures and VA FileMan's programmer API. It is assumed that the user of the tutorial has full, unrestricted programmer access in an M test account.

Objectives

After completing this tutorial, you should have the skills to create a ScreenMan interface to VA FileMan files. You will be able to:

- Create a ScreenMan Form.
- Add read-only fields to the form.
- Add editable fields to the form.
- Use the form editor to visually position fields on blocks.
- Add additional pages to the form.

- Edit multiples on the form (two methods).
- Add ScreenMan computed fields and form-only fields to the form.
- Edit fields in another file's related entries.
- Use branching logic to jump from one field to another.
- Turn the ScreenMan form into a Menu Manager option.

Files Used

This tutorial needs the following files to be present in your test account:

- NEW PERSON (primary file used by tutorial; you add a ScreenMan form to it.)
- OPTION (your form will navigate to edit fields in this file.)
- SERVICE/SECTION (your form will navigate to edit fields in this file.)

Advance Preparation

This is a hands-on tutorial, requiring a safe (that is, non-production) M account in which you can build a ScreenMan form.

The following software must be installed in the test account you use for the tutorial:

- VA FileMan V. 21.0 or VA FileMan V. 22.0.
- Kernel V. 7.1 or Kernel V. 8.0.

Kernel provides access to the ScreenMan options, as well as providing the files for which you'll build a ScreenMan form during the tutorial (see above).

You need to be granted the following privileges in the test M account you use for the tutorial:

- The XUSCREENMAN security key.
- A File Manager Access Code of "@".

It is further assumed that you have full, unrestricted programmer access in this test account, including the ability to run any option and to add options to your menu tree without supervision.

Each tutorial user should choose a unique namespace beginning with Z, concatenated with two or three other letters, for example ZYXU. Use this namespace as the beginning of the names for all forms and blocks that you create during this tutorial. Using your namespace will protect the system you are using from having existing forms and blocks overwritten. This namespace is referred to as Zxxx during the tutorial.

Introduction

1. Starting Out with Forms

This unit of the tutorial shows you how to start building a form. It demonstrates the initial building of a form, how to add blocks to a form, and how to add simple fields to a block. In this section of the tutorial, you will:

- Create your first form.
- Add a Display block to the form.
- Add a caption field to the Display block.
- Add (read-only) fields from the NEW PERSON file to the Display block on the form.
- Run the form.
- Add an Edit block to the form.
- Add (editable) fields from the NEW PERSON file to the Edit block on the form.
- List the form characteristics using the PRINT^DDS direct-mode utility.

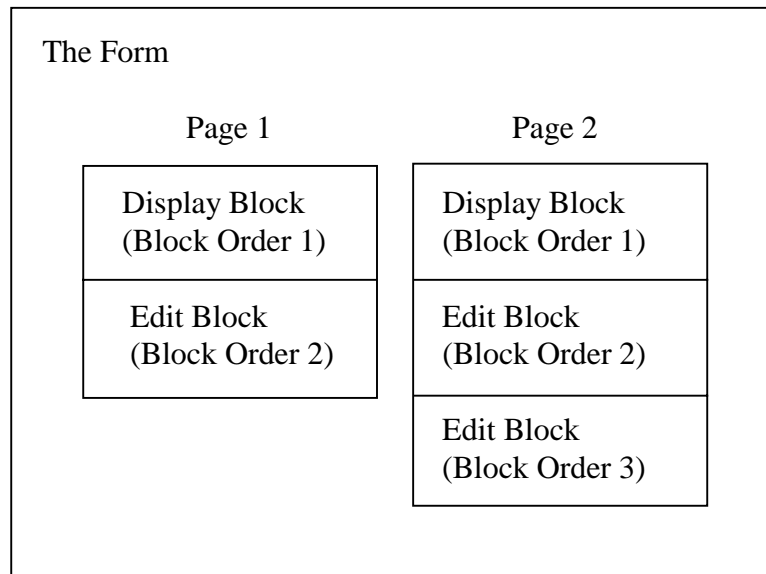
1-1. Create a Form

ScreenMan is a tool for screen-oriented, form-oriented data entry into VA FileMan files. The Form Editor tool was introduced with VA FileMan V. 21.0. It assists developers in creating ScreenMan interfaces. This tutorial concentrates on creating forms with the Form Editor.

In this tutorial, you will create a form to edit entries in the NEW PERSON file. You won't edit every field in the NEW PERSON file (that could be a big form!). You will gradually add selected fields from the NEW PERSON file to the form, in such a way that you get experience using a variety of the features ScreenMan. At the end of the tutorial, you will have built a form that is very similar to Kernel's Edit User Characteristics option (although your form will edit a different set of fields).

First, let's discuss ScreenMan forms themselves. When you create a form, a first page is created for the form by default. But you cannot place fields directly on either a form or page; you must create blocks first. This is because the only place fields can be placed on is a block. A block provides a place to logically group a set of fields—you might edit a multiple's fields in one block on one page, and the top-level fields for the file on a block on another page. You can also designate whether the fields on any given block are display-only or editable.

The following diagram shows the relationship between blocks, pages, and forms:



Thus, to create a form, the first step is to create the form itself. The first page of the form is created automatically when the form is created. However, before you can

add fields to the form, you must add at least one block to the form. Adding a block to the form you create will be your second step.

The most basic element in a ScreenMan form is the form itself. To create the form, you will use a tool called the Form Editor. You can invoke the Form Editor from the Edit/Create a Form option. The menu path to this option is:

```
VA FileMan
  Other Options
    ScreenMan
      Edit/Create a Form
```

Alternatively, you can invoke the Form Editor from programmer mode using the direct-mode utility `D ^DDGF`.

► **To create your first form:**

1. Invoke the Form Editor, using either the menu option or the direct-mode utility.
2. The first thing you are asked is for what file to create a form. Since we are creating a form to edit the NEW PERSON file, enter NEW PERSON.
3. You are now asked for a form name. Entering two question marks lists current forms defined for the NEW PERSON file. Enter a new unique form name using your student namespace concatenated with "EDIT". For example, if your namespace is ZYXU, add a form called ZYXU EDIT.

NOTE: From hereon in, the namespace part of a name will be referred to as "Zxxx" in the tutorial.

```
Select ScreenMan Option: Edit/Create a Form <RET>

EDIT/CREATE FORM FOR WHAT FILE: NEW PERSON <RET>

Select FORM: Zxxx EDIT <RET>
Are you adding 'Zxxx EDIT' as a new FORM? Y <RET>
```

4. Once you select a name for the form, you are placed in the ScreenMan's Form Editor. In subsequent tutorial sections, you will build on this form using the Form Editor, gradually creating a robust interface to the NEW PERSON file.
5. Press `<PF1>E` to save the new form and exit the Form Editor.

In the next lesson, you will add a DISPLAY block to your new form.

Summary

To create a ScreenMan form, use the Form Editor. The Form Editor can be invoked with the Edit/Create a Form option, and also with the direct-mode utility D ^DDGF.

1-2. Add a DISPLAY Block to the Form

The new form you created in the first lesson has a page (page 1 is created by default), but no blocks yet. In order to add any fields to the form, you need to place a block on the form. Thus, the first thing to do with your new form is to add a block.

There are two types of blocks, EDIT (which allows fields to be changed by the user), and DISPLAY (which allows fields to be displayed but not changed by the user).

Your cursor position, at the time you add a block, defines the region of the block. The region of the page covered by the block starts at your cursor position, and moves down and to the right. Thus, if your block covers only the bottom half of the page, you will only be able to place fields on the bottom half of that page (for that block only).

You can add more than one block on a page. The following are some reasons why you might use more than one block on a page:

- Reusing a block. You may have a multi-page form, for example, and want the same header information on each page of the form. You can create a block for the header information only, and reuse the same block on each page of the form. Each page would then need an additional block for the fields unique to each page.
- If fields on the same page come from different data dictionary file numbers (either from a multiple, or from another file), you need to use separate blocks. You can only place fields from a single data dictionary file number on any given block.

NOTE: Both of these situations are covered in subsequent lessons in this tutorial.


- You can make a field on an EDIT block "display-only" by individually setting that field's DISABLE EDITING property. You may have a large number of display-only fields, as well as some fields to edit, however. In this case it may be easier to place the display-only fields on their own DISPLAY block, rather than set each field's DISABLE EDITING property.

In this lesson, we'll add a single block to the page, and make it a DISPLAY block.

► **To add a display block:**

1. In the Form Editor, with your Zxxx EDIT form loaded, use the arrow keys to position the cursor at row 1, column 1 if it is not already at that position. This is the top left corner on the form.

Hot Tip



Cursor Position

The current cursor position is always displayed at the lower right corner of the Form Editor screen.

2. At the current cursor position (row 1, column 1) add a block to the form at the cursor by pressing **<PF2>**.
3. You're asked to select a new block name. Enter your namespace, concatenated with "MAIN 1" (e.g., "**Zxxx MAIN 1**").
4. At the "Are you adding Zxxx MAIN 1 as a new block on this page?" prompt, answer **YES**.
5. You are now given a chance to enter the properties for the Zxxx MAIN 1 block. Change the TYPE OF BLOCK to DISPLAY. The BLOCK ORDER should be 1. Accept the other defaults. The block properties should match those shown in the picture that follows:

Block Properties Stored in FORM File	
BLOCK NAME: Zxxx MAIN 1	BLOCK ORDER: 1
TYPE OF BLOCK: DISPLAY	OTHER PARAMETERS...
POINTER LINK:	
PRE ACTION:	
POST ACTION:	
Block Properties Stored in BLOCK File	
NAME: Zxxx MAIN 1	DESCRIPTION (WP):
DD NUMBER: 200	DISABLE NAVIGATION:
PRE ACTION:	
POST ACTION:	

6. Press **<PF1><E>** to save the new block.

Any fields placed on this block will be read-only, since this is a DISPLAY block; in the next lesson, you will add several fields to the block.

Hot Tip



Seeing Blocks, and Editing Block Properties

How do you see what blocks are on a page? Easy - switch to the Block Viewer screen (press <PF1><V>). The name of each block is displayed as a caption on the page.

You can move the position of a block by selecting it (move the cursor to the label of the block, and <SPACE> or <RET>), moving it with the arrow keys, and deselecting it (press <SPACE> or <RET> again).

You can edit the properties of the block by moving the cursor to the block's caption, selecting it (pressing <SPACE> or <RET>), and pressing <PF4> to edit the properties of the block.

When you are done working with blocks, and want to exit the Block Viewer screen, press <PF1><V> again to switch back to the main Form Editor screen.

Summary

To add an edit or display block to a form, press <PF2> and select the block properties. The position your cursor is at when you press <PF2> defines the regions of the block. You need to add at least one block to every page, because fields must be placed on blocks (not directly on the page).

1-3. Add a Title (Caption-Only Field)

Now that you have added a block to the form, you can start adding fields to the block. First let's add a caption to the block, displaying the title "NEW PERSON EDIT" at the top of the form.

► **To add a caption-only field to the block:**

1. Using the arrow keys, position the cursor at row 1, column 1.
2. At the current cursor position, press <PF2><F>. This brings up a dialogue similar to the following:

```
Select BLOCK: Zxxx MAIN 1
FIELD ORDER: 1
FIELD TYPE: DATA DICTIONARY FIELD
```

Accept the defaults for BLOCK and FIELD ORDER. **Change** the FIELD TYPE to **CAPTION ONLY**. Save the settings (press <PF1><E> to save and exit this dialogue.)

```
Select BLOCK: Zxxx MAIN 1
FIELD ORDER: 1
FIELD TYPE: CAPTION ONLY
```

3. Once you save the initial field settings, the Form Editor presents the Caption-Only Field Properties window. Set the properties as follows:

```

Caption-Only Field Properties

FIELD ORDER: 1
CAPTION: NEW PERSON EDIT
UNIQUE NAME:

EXECUTABLE CAPTION:
CAPTION COORDINATE: 1,1
```

4. Press <PF1><E> to save the field properties and exit this dialogue.

Your form should now have the caption "NEW PERSON EDIT", left-justified on the first line.

Hot Tip



Re-Editing Field Properties

To re-edit the properties of a field, move the cursor to the field's *caption*, press **<SPACE>** or **<RET>** to select the field, and press **<PF4>** to edit the properties.

Summary

You can add fields to a block by pressing **<PF2><F>** and filling in the properties. To add a caption to a block, choose a field type of **CAPTION ONLY**.

1-4. Re-position Fields on a Block

In the previous lesson, we added a caption field to our first block. Now let's use the Form Editor's visual editing capabilities to re-position the caption field on our block. To do this, first you need to select the field. Then you can move it. Finally, you will de-select the field.

► To select the field:

1. In the Form Editor, use the arrow keys to position the cursor at the first letter in the *caption* NEW PERSON EDIT.
2. Press <SPACE> or <RET> with the cursor in the field caption to select the field; this will highlight the entire field.

► To move the field:

1. With the field highlighted, move it by pressing the **arrow keys**. Move the field to the center of row 1. The leftmost character of the field should be at row 1, column 32; you can tell by the R1,C32 indicator in the lower-right corner of the window.

NEW PERSON EDIT	
File: NEW PERSON (#200)	R1,C32
Form: Zxxx EDIT	
Page: 1 (Page 1)	
<PF1>Q=Quit <PF1>E=Exit <PF1>S=Save <PF1>V=Block Viewer <PF1>H=Help	

► **To de-select the field:**

1. With the field highlighted (and in position) press <SPACE> or <RET> to de-select (and drop) the field into place.

Hot Tip



Moving Screen Elements

There are other shortcut keystrokes that you can use when moving screen elements such as fields. These keystrokes (move field to top of screen, move field to bottom of screen, etc.) are described in the "ScreenMan" chapter in the *VA FileMan Getting Started Manual* (V. 22.0).

Summary

The Form Editor provides visual field positioning capabilities. This lets you quickly position fields on blocks, which helps you build visually attractive forms very quickly.

1-5. Add Data Dictionary Fields

At this point, we've added one field, a caption-only field, to our first block. Now let's add some data dictionary fields to the display-only block from the NEW PERSON file: NAME and SSN. Because these fields will be placed on a display-only block, the values will be displayed to the form user, but won't be editable.

► To add the NAME field:

1. To add the NAME field, position the cursor at the beginning of the third line on the screen, using the arrow keys, and press <PF2><F> to add a field to the block. Set the field up as follows:

```
Select BLOCK: Zxxx MAIN 1
FIELD ORDER: 2
FIELD TYPE: DATA DICTIONARY FIELD
```

Press <PF1><E> to save these field settings.

2. The Form Editor presents the Data Dictionary Field Properties window next. For FIELD, choose **NAME**.
3. Press <PF1><E> to save the field properties. The Form Editor returns you to the form screen, and you should see the NAME field now on the screen:

```

NEW PERSON EDIT

NAME: _____

File: NEW PERSON (#200)                                     R3,C1
Form: Zxxx EDIT
Page: 1 (Page 1)

<PF1>Q=Quit  <PF1>E=Exit  <PF1>S=Save  <PF1>V=Block Viewer  <PF1>H=Help

```

► **To add the SSN field:**

1. To add the SSN field, position the cursor on the middle of the third line, and press **<PF2> <F>** to add the SSN field. Set it up as follows:

```
Select BLOCK: Zxxx MAIN 1
FIELD ORDER: 3
FIELD TYPE: DATA DICTIONARY FIELD
```

Press **<PF1><E>** to save these field settings.

2. The Form Editor presents the Data Dictionary Field Properties window next. For FIELD, choose **SSN**.
3. Press **<PF1><E>** to save the field properties. The Form Editor returns you to the form screen. You should see the SSN field now on the block:

```
NEW PERSON EDIT
NAME: _____ SSN: _____

File: NEW PERSON (#200) R3,C39
Form: Zxxx EDIT
Page: 1 (Page 1)

<PF1>Q=Quit <PF1>E=Exit <PF1>S=Save <PF1>V=Block Viewer <PF1>H=Help
```

If you need to, you can re-position the fields in the same way that you re-positioned the "NEW PERSON EDIT" caption previously.

► **To reposition fields:**

1. Select a field (move the cursor to the field **caption**, and press **<SPACE>** or **<RET>**.)
2. Use the arrow keys to move the field.
3. Press **<RET>** to drop the field in its new location.

Hot Tip**Positioning Data Dictionary Fields: Caption Portion and Data Field Portion**

Data dictionary fields have two parts, caption and data field, which can be positioned independently. If you position the cursor under the caption of a data dictionary field and press **<SPACE>** or **<RET>**, you select the caption and data field parts together. However, if you position the cursor under the data field (the underlined portion) and then select, you select the data field only, and can position it independently of the field caption. In this tutorial, you should always move the caption and data field together.

Summary

Adding data dictionary fields is similar to adding caption-only fields; you simply need to choose what field you want to add.

1-6. Running the Form

At this point, we've added one block and three fields to our form. To see what the form looks like so far from a user's point of view, use the VA FileMan option Run a Form (on the same menu as the Edit/Create a Form option).

First, in the Form Editor screen, press **<PF1><E>** to save the form you've been working on and exit. Next, choose the **Run a Form** option. Run a form from the NEW PERSON file, and choose the form you've been building:

```
Select ScreenMan Option: Run a Form <RET>
RUN FORM FROM WHAT FILE: NEW PERSON// <RET>
Select FORM: Zxxx EDIT <RET>
Enter number of first page: 1// <RET>
Select NEW PERSON name: SMITH,JANE <RET>
```

Your form should look something like this (notice that you can't edit the NAME or SSN fields, which reside on a DISPLAY block):

NEW PERSON EDIT	
NAME: SMITH,JANE	SSN: 890234123
Exit Save Refresh	
Enter a command or '^' followed by a caption to jump to a specific field.	
COMMAND:	Press <PF1>H for help
<div style="background-color: black; width: 100px; height: 15px; display: inline-block;"></div> INSERT	

Summary

Use the Run a Form option to test out your forms.

1-7. Add an Edit Block to the Form

In previous lessons, we added fields to a DISPLAY-type block. Because of the block they were added to, these fields are read-only. In this lesson, we'll add an EDIT-type block to our form; this will let us add fields to the form that the user can actually edit.

► To add the edit block:

1. Invoke the Form Editor again, and load the Zxxx EDIT form.
2. In the Form Editor, use the arrow keys to position the cursor at row 5, column 1 on the form.
3. Add a block to the form at the cursor position (row 5, column 1) by pressing **<PF2>**.
4. You are asked to select a new block name. Enter your namespace, concatenated with "MAIN 2" (e.g., "**Zxxx MAIN 2**").
5. At the "Are you adding Zxxx MAIN 2 as a new block?" prompt, answer **YES**.
6. At the "Are you adding Zxxx MAIN 2 as a new block on this page?" prompt, answer **YES**. You are now given a chance to enter the properties for the Zxxx MAIN 2 block.
7. In the Properties window, the Type for this block should be **EDIT**, and the **BLOCK ORDER** should be **2**; accept the other defaults. Press **<PF1><E>** to save the new block and exit.

Block Properties Stored in FORM File	
BLOCK NAME: Zxxx MAIN 2	BLOCK ORDER: 2
TYPE OF BLOCK: EDIT	OTHER PARAMETERS...
POINTER LINK:	
PRE ACTION:	
POST ACTION:	
Block Properties Stored in BLOCK File	
NAME: Zxxx MAIN 2	DESCRIPTION (WP):
DD NUMBER: 200	DISABLE NAVIGATION:
PRE ACTION:	
POST ACTION:	

Hot Tip



Editing Block Properties at a Later Time

To edit a block's properties at a later time:

1. Press **<PF1><V>** to go to the Block Viewer screen.
2. Select the block whose properties you want to edit by positioning the cursor over it and pressing **<SPACE>** or **<RET>**.
3. Press **<PF4>** to edit the block's properties in the Block Properties window. Once you are done editing properties, save the changes and exit by pressing **<PF1><E>**, or quit without saving by pressing **<PF1><Q>**.
4. Once you are back in the Block Viewer screen, press **<PF1><V>** to return to the normal Form Editor screen.

To delete a block, edit the block's properties, and delete the block by entering an at-sign ("@") in the BLOCK NAME field.

Summary

Adding an EDIT block to a form enables the editing of any data dictionary fields placed on the EDIT-type block.

1-8. Add Data Dictionary Fields to the EDIT Block

In the previous lesson, we added an EDIT-type block to our form. Now that we've done that, we can add some data dictionary fields to the EDIT block. This means that the user will actually be able to edit data in the NEW PERSON file when they use the form.

In this lesson, we'll add the following fields to the edit block (Zxxx MAIN 2), in the order listed below:

1. PHONE
2. PHONE #3
3. PHONE #4
4. OFFICE PHONE
5. AUTO MENU
6. INITIAL
7. NICK NAME
8. PREFERRED EDITOR
9. TYPE-AHEAD

► To add each field:

1. Position the cursor somewhere on the Zxxx MAIN 2 block (which is anywhere from row 5 to the bottom of the form), roughly where you would like to place the field.
2. Press <PF2><F> to add the individual field.
3. When the Form Editor asks for block, make sure you choose **Zxxx MAIN 2**. Accept the default field order, and make sure the field type is Data Dictionary:

```
Select BLOCK: Zxxx MAIN 2
FIELD ORDER: 1
FIELD TYPE: DATA DICTIONARY FIELD
```

4. Press <PF1><E> to save and exit this dialogue.
5. When you reach the Field Properties window, choose the appropriate field by entering the field name in the FIELD property. Then press <PF1><E> to save the new field.

► **To change the field order for any field:**

1. Select the field whose order you want to change (move the cursor to the field **caption**, and press <SPACE> or <RET>).
2. Press <PF4> to edit the field properties.
3. Change the field order to a different number (decimal fractions are allowed).

Alternatively, you can reset the field order of *all* fields to be left-to-right, top-to-bottom.

► **To reset the field order of all fields to left-to-right, top-to-bottom:**

1. From the Form Editor, go to the Block Viewer screen by pressing <PF1><V>.
2. Select the block in question (move cursor to it, and press <SPACE> or <RET>).
3. Press <PF1><O> to reorder the field sequence.
4. De-select the block (by pressing <SPACE> or <RET> again).
5. Press <PF1><V> to return to the normal Form Editor screen.

Congratulations! Making an interface to edit fields in the NEW PERSON file was as easy as that!

Hot Tip



Display Width vs. Actual Width for Fields

Some fields are longer than a form is wide; fields can be up to 245 characters in length. How can you edit a field like this in a ScreenMan form, which is only 80 characters wide? The answer is that when you place a field like this on a form, you can set the field's display width by setting its DATA LENGTH property. You could set its DATA LENGTH to be 75, for example. Then, when the user runs the form, the field will be displayed in a 75-character-wide window. To see and edit the entire field, the user can press <PF1><Z>; this opens a large edit area at the bottom of the screen.

You can use this technique even with fields that fit within the width of your form. For example, you might have a field that is 60 characters in length, but set it so that it only takes up 30 characters when displayed on your form (by setting its DATA LENGTH to 30).

Summary

To let users edit data in a VA FileMan file, you can add data dictionary fields to an Edit-type block on the form. The field order (when the user traverses fields by pressing <RET>) is, by default, the order you add fields to the form. You can change the field order at a later time, if necessary.

1-9. Use PRINT^DDS to List Form Characteristics

At this point, we've added several blocks and a number of fields to our form.

When you are working with forms, you may want to generate a report on your form so that you can review your form's characteristics. ScreenMan provides a way to do this; you can use the direct-mode utility D PRINT^DDS to list all characteristics of a form. It lists the properties of the form and of all blocks and fields used on the form.

From the M programmer prompt, run PRINT^DDS and compare the output for your form to the output listed below. The printout for your form thus far should correspond to the printout listed below:

```

FORM LISTING - Zxxx EDIT (#52)
FILE: NEW PERSON (#200)                                AUG 09, 1995  08:18    PAGE 1
-----
      PRIMARY FILE: 200                                READ ACCESS: @
      DATE CREATED: AUG 08, 1995@13:26                WRITE ACCESS: @
      DATE LAST USED: AUG 09, 1995@08:18              CREATOR: 34

Page  Page
Number Properties
-----
1      Page 1

PAGE COORDINATE:          1,1

Block  Block
Order  Properties (Form File)
-----
1      Zxxx MAIN 1 (#152)

      TYPE OF BLOCK:          DISPLAY
      BLOCK COORDINATE:      1,1

Block Properties (Block File)
-----
DATA DICTIONARY NUMBER: 200

Field  Field
Order  Properties
-----
1      FIELD TYPE:          CAPTION ONLY
      CAPTION:              NEW PERSON EDIT
      CAPTION COORDINATE:   1,32

2      FIELD TYPE:          DATA DICTIONARY FIELD
      CAPTION:              NAME
      FIELD:                .01
      CAPTION COORDINATE:   3,1
      DATA COORDINATE:     3,7
      DATA LENGTH:        30

```

Starting Out with Forms

```
3      FIELD TYPE:          DATA DICTIONARY FIELD
      CAPTION:              SSN
      FIELD:                9
      CAPTION COORDINATE:   3,45
      DATA COORDINATE:     3,50
      DATA LENGTH:        9

2      Zxxx MAIN 2 (#153)

      TYPE OF BLOCK:        EDIT
      BLOCK COORDINATE:     5,1

      Block Properties (Block File)
      -----
      DATA DICTIONARY NUMBER: 200

      Field  Field
      Order  Properties
      -----
1      FIELD TYPE:          DATA DICTIONARY FIELD
      CAPTION:              PHONE
      FIELD:                .131
      CAPTION COORDINATE:   2,14
      DATA COORDINATE:     2,21
      DATA LENGTH:        20

2      FIELD TYPE:          DATA DICTIONARY FIELD
      CAPTION:              PHONE #3
      FIELD:                .133
      CAPTION COORDINATE:   3,11
      DATA COORDINATE:     3,21
      DATA LENGTH:        20

3      FIELD TYPE:          DATA DICTIONARY FIELD
      CAPTION:              PHONE #4
      FIELD:                .134
      CAPTION COORDINATE:   4,11
      DATA COORDINATE:     4,21
      DATA LENGTH:        20

4      FIELD TYPE:          DATA DICTIONARY FIELD
      CAPTION:              OFFICE PHONE
      FIELD:                .132
      CAPTION COORDINATE:   5,7
      DATA COORDINATE:     5,21
      DATA LENGTH:        20

5      FIELD TYPE:          DATA DICTIONARY FIELD
      CAPTION:              AUTO MENU
      FIELD:                200.06
      CAPTION COORDINATE:   6,10
      DATA COORDINATE:     6,21
      DATA LENGTH:        20

6      FIELD TYPE:          DATA DICTIONARY FIELD
      CAPTION:              INITIAL
      FIELD:                1
      CAPTION COORDINATE:   7,12
      DATA COORDINATE:     7,21
      DATA LENGTH:        5

7      FIELD TYPE:          DATA DICTIONARY FIELD
      CAPTION:              NICK NAME
      FIELD:                13
      CAPTION COORDINATE:   8,10
      DATA COORDINATE:     8,21
      DATA LENGTH:        10
```

```

8      FIELD TYPE:          DATA DICTIONARY FIELD
      CAPTION:             PREFERRED EDITOR
      FIELD:                31.3
      CAPTION COORDINATE:  9,3
      DATA COORDINATE:    9,21
      DATA LENGTH:        30

9      FIELD TYPE:          DATA DICTIONARY FIELD
      CAPTION:             TYPE-AHEAD
      FIELD:                200.09
      CAPTION COORDINATE:  10,9
      DATA COORDINATE:    10,21
      DATA LENGTH:        11

```

Summary

Use PRINT^DDS to list the characteristics of a form.

- This is the end of the first section of the hands-on tutorial. Congratulations! You have created a ScreenMan form, linked it to a file, and added both display-only and editable fields to the form. If you want to try out some of the more advanced features of ScreenMan, continue on to the next section of this tutorial!

2. Pages, Multiples, and Computed Fields

This unit of the tutorial demonstrates some of the more typical things a developer might want to do with a form, beyond the adding of simple fields to a single page. It shows how to add pages to a form, and shows two ways of editing multiple fields on a form. Also, it shows how to enhance the presentation of information with ScreenMan computed fields. You will:

- Add a second page to your form.
- Edit a multiple field using a repeating block.
- Edit a multiple field using a pop-up page.
- Resize and reposition the pop-up page.
- Add a ScreenMan computed field, enhancing the presentation of the multiple in the repeating block.

2-1. Add a Second Page

Forms can have more than one page. They can have many pages, each page with its own blocks. Some reasons to use more than one page in a form are:

- There are too many fields to fit on one page
- You want to edit fields in a different file or in a multiple
- You want to display information in a "pop-up" page

The general process to add a page to a form is as follows:

1. Press <PF2><P> to add a page.
2. Choose a unique page number for the page.
3. In the new page's properties, fill in a page number for both the NEXT PAGE and PREVIOUS PAGE properties. This tells ScreenMan what pages to go to when the user moves backwards or forwards from the page.
4. Adjust the PREVIOUS PAGE and NEXT PAGE properties of other pages to make sure that the total sequence of pages is adjusted to include your new page.
5. Add a block to the new page. Every page should have at least one block.

Let's add a second page to the form we've been working on.

► To add a page to the form:

1. In the Form Editor, load the Zxxx EDIT USER form.
2. Add a page by pressing <PF2><P>.
3. In a pop-up window, you are asked for the NEW PAGE NUMBER. In general, you should number the pages in your form sequentially, so it is easy to keep track of them. Since this will be the second page in the form, enter **2**.
4. Answer **YES** to the "Are you adding Page 2 as a new page on this form?" prompt.

5. The Form Editor presents a "Page Properties" window letting you edit the characteristics of the new page. Enter **1** for the PREVIOUS PAGE and NEXT PAGE properties; accept the defaults for all other properties. The page properties should match those in the picture below:

```
Page Properties

PAGE NUMBER: 2
PAGE NAME: Page 2
HEADER BLOCK:

PAGE COORDINATE: 1,1
IS THIS A POP UP PAGE?
LOWER RIGHT COORDINATE:

NEXT PAGE: 1
PREVIOUS PAGE: 1
PARENT FIELD:

DESCRIPTION (WP):
PRE ACTION:
POST ACTION:
```

6. Press **<PF1><E>** to save the new page. The Form Editor now displays the blank second page of the form. Next we'll need to add a block to the page.

The new page needs at least one block. Let's add an edit-type block to the page.

► **To add an edit block:**

1. In the Form Editor, on page 2, position the cursor at row 1, column 1 (top left corner).
2. Add a block to the form at the cursor position (row 1, column 1) by pressing **<PF2>**.
3. You are asked to select a new block name. Enter your namespace, concatenated with "MAIN 3" (e.g., "**Zxxx MAIN 3**").
4. At the "Are you adding Zxxx MAIN 3 as a new block on this page?" prompt, answer **YES**. You are now given a chance to enter the properties for the Zxxx MAIN 3 block.

5. The Type for this block should be EDIT, and the BLOCK ORDER should be 1; accept the other defaults. The block properties should match those shown in the picture below:

Block Properties Stored in FORM File	
BLOCK NAME: Zxxx MAIN 3	BLOCK ORDER: 1
TYPE OF BLOCK: EDIT	OTHER PARAMETERS...
POINTER LINK:	
PRE ACTION:	
POST ACTION:	
Block Properties Stored in BLOCK File	
NAME: Zxxx MAIN 3	DESCRIPTION (WP):
DD NUMBER: 200	DISABLE NAVIGATION:
PRE ACTION:	
POST ACTION:	

6. Press **<PF1><E>** to save the new block.

Hot Tip



Shortcuts to Move to Another Page

For other shortcuts besides **<PF1><P>** for navigating from one page to another, see the "ScreenMan" chapter in the *VA FileMan Getting Started Manual (V. 22.0)*.

Now, let's link the first page to the second. Do this by moving back to the first page and setting the first page's NEXT PAGE property to 2.

► To link the first page to the second:

1. Press **<PF1><P>**, to change pages, and enter **1** as the page to change to.
2. Once page 1 is the current page, enter **<PF4><P>** to edit the first page's properties.
3. For the NEXT PAGE and PREVIOUS PAGE properties, enter **2**.
4. Press **<PF1><E>** to save the properties for the first page.

Once you're back in the Form Editor, press **<PF1><E>** to save the form and exit. Test the form using the Run a Form option. You should have a new choice, NEXT

Pages, Multiples, and Computed Fields

PAGE, from the first page at the COMMAND prompt. Choosing **NEXT PAGE** at the COMMAND prompt should move you to the (currently blank) second page of the form. Also, <PF1><ARROW-DOWN> should move you to the next page, and <PF1><ARROW-UP> should move you to the previous page.

As a final touch, you may want to add two CAPTION fields at the top right corners of the two pages: "Page 1 of 2" on the first page, and "Page 2 of 2" on the second page.

```
Page Properties

PAGE NUMBER: 1
PAGE NAME: Page 1
HEADER BLOCK:

PAGE COORDINATE: 1,1
IS THIS A POP UP PAGE?
LOWER RIGHT COORDINATE:

NEXT PAGE: 2
PREVIOUS PAGE: 2
PARENT FIELD:

DESCRIPTION (WP):
PRE ACTION:
POST ACTION:
```

Summary

Forms can have multiple pages. When you add pages, your main task is to make sure that your pages all have appropriate PREVIOUS PAGE and NEXT PAGE links so that a correct sequence of pages is available to users of your form.

2-2. Add a Multiple Field (in a Repeating Block)

You may want the user to be able to edit data that is in a multiple. The two common ways to edit multiples on forms are:

- Add a repeating block that is dedicated to the multiple
- Add a pop-up page that is dedicated to the multiple

In this lesson, we'll let the user edit data in a multiple using a repeating block. We'll add the ability on our form to edit the SECONDARY MENU OPTIONS multiple in the NEW PERSON file.

A repeating block presents multiple entries in a scrolling list, making it possible to display more than one entry from the multiple to the user at the same time. The main limitation to using the repeating block is that all fields to be edited must fit on one line on the screen.

The general process to set up a multiple in a repeating block is:

1. Add a new block (of type EDIT) to a page.
2. In the new block's OTHER PARAMETERS section, set the REPLICATION property to the length of the list to display multiple entries in.
3. Set the DD NUMBER property of the new block to the DD number of the multiple.
4. Add the fields to edit in the multiple, all in one line, to the repeating block (remove all captions).
5. If you want to add column "heading" captions above the row of fields, you can do so, but add the captions to a different block (not the repeating block). The only fields that should be on the repeating block should be one line of data fields.

We'll add a repeating block to the second page of the form to edit the SECONDARY MENU OPTIONS multiple in the NEW PERSON file.

► **To add a repeating block:**

1. Position the cursor on the second page of the form, at row 5, column 7.
2. Press **<PF2>** to add a new block to the current page. For block name, enter **Zxxx REPEATING 1**.
3. Answer **YES** to the "Are you adding Zxxx REPEATING 1 as a new block on this page?" prompt.
4. The TYPE for this block should be **EDIT**, and the BLOCK ORDER should be **2**.
5. Under OTHER PARAMETERS, set REPLICATION to be **4**. This makes the block a repeating block and sets the number of entries to display on screen to **4**. Press **<PF1><C>** to close the OTHER PARAMETERS window.
6. Set the DD NUMBER block property to be **200.03** (the data dictionary number of the SECONDARY MENU OPTIONS multiple).

Block Properties Stored in FORM File	
BLOCK NAME: Zxxx REPEATING 1	BLOCK ORDER: 2
TYPE OF BLOCK: EDIT	OTHER PARAMETERS...
POINTER LINK:	
PRE ACTION:	
POST ACTION:	

Block Properties Stored in BLOCK File	
NAME: Zxxx REPEATING 1	DESCRIPTION (WP):
DD NUMBER: 200.03	DISABLE NAVIGATION:
PRE ACTION:	
POST ACTION:	

7. Save the new block by pressing **<PF1><E>** on its Block Properties page.

Now lets add the fields to the repeating block that we want users to be able to edit. Remember, we can only add fields that will fit on one line to a repeating block. We'll add two fields on one line: the .01 field and the SYNONYM field.

► **To add the SECONDARY MENU OPTIONS .01 field to the repeating block:**

1. Position the cursor on the second page of the form, at row 5, column 7.
2. Press <PF2><F> to add a field at the cursor position. Add the field to the Zxxx REPEATING 1 block, with a field type of DATA DICTIONARY FIELD.

```
Select BLOCK: Zxxx REPEATING 1
FIELD ORDER: 1
FIELD TYPE: DATA DICTIONARY FIELD
```

3. On the Data Dictionary Field Properties screen for the new field, set the FIELD to **SECONDARY MENU OPTIONS** (# .01). You'll notice that the only fields selectable are those in the SECONDARY MENU OPTIONS multiple (because we set the block to DD NUMBER 200.03).
4. Delete the text in the CAPTION property.
5. Press <PF1><E> to save the field properties. You should see an underlined field on the Form Editor screen, with no caption.

► **To add the SECONDARY MENU OPTIONS (SYNONYM) field to the repeating block:**

1. Position the cursor on the second page of the form, at row 5, column 40.
2. Press <PF2><F> to add a field at the cursor position. Add the field to the Zxxx REPEATING 1 block, with a field type of DATA DICTIONARY FIELD.

```
Select BLOCK: Zxxx REPEATING 1
FIELD ORDER: 2
FIELD TYPE: DATA DICTIONARY FIELD
```

3. On the Data Dictionary Field Properties screen for the new field, set the FIELD to **SYNONYM** (#2).
4. Delete the text in the CAPTION property.
5. Press <PF1><E> to save the field properties. You should see a second underlined field on the Form Editor screen, with no caption.

Finally, add four caption-only fields just above the repeating block to provide neat columnar headings for the repeating block fields below.

► **To add caption-only field columnar headings to the repeating block:**

1. Position the cursor at row 3, column 7, and add a caption-only field to the Zxxx MAIN 3 block (**not the repeating block!**). Make sure you set the FIELD TYPE to CAPTION ONLY:

```
Select BLOCK: Zxxx MAIN 3
FIELD ORDER: 1
FIELD TYPE: CAPTION ONLY
```

On the Caption Only Field Properties screen, set the CAPTION to **"Secondary Menu Option"**.

2. Position the cursor at row 3, column 40, add a caption-only field to the Zxxx MAIN 3 block, and set the CAPTION to **"Synonym"**.
3. Position the cursor at row 4, column 7, and add a caption-only field to the Zxxx MAIN 3 block. At the CAPTION property, enter:

```
!DUP("-" , 30)
```

This is a shortcut way of making a caption of 30 hyphens.

4. Position the cursor at row 4, column 40, add a caption-only field to the Zxxx MAIN 3 block, and set the CAPTION to "-----".

If you don't get the captions to line up exactly, select the caption-only field in question (move cursor to the field and press <SPACE> or <RET>), use the arrow keys to position the caption, and press <SPACE> or <RET> again to de-select the field.

Now exit the Form Editor (<PF1><E>) and run the form. Press <PF1><ARROW-DOWN>, <NEXT SCREEN>, or <PAGE DOWN> to go to page 2. There are four display lines for the SECONDARY MENU OPTIONS multiple. Below the last item in the list, try entering some options, for example, "XU FINDUSER", "XU FIRST LINE PRINT", etc. Notice that you can scroll up and down the list. Enter more than four options in the multiple so that you can see how the scrolling works.

Page 2 of 2

Secondary Menu Option	Synonym
XU FINDUSER	
XU FIRST LINE PRINT	
XUUSERDISP	
+DIUSER	

Exit Save Refresh

Enter a command or '^' followed by a caption to jump to a specific field.

COMMAND: XXXXXXXXXX Press <PF1>H for help INSERT

Hot Tip



Caption Prompt Shortcuts

Other shortcuts at the caption prompt are listed in the "ScreenMan" chapter in the *VA FileMan Getting Started Manual (V. 22.0)*.

Summary

You can use a repeating block to edit multiples. The advantage of repeating blocks is that you can present a list of entries from a multiple on your form; the disadvantage is that all the fields from the multiple that you want to display or edit must fit on a single line.

2-3. Add a Multiple Field (in a Pop-Up Page)

In the previous lesson, we used a repeating block to let the user edit one line's worth of fields from a multiple.

You may want the user to be able to edit many fields from a given a multiple, however. You can provide this ability by creating a pop-up page, and setting up the multiple to be edited there.

The general process to do set up a multiple to be edited in a pop-up page is:

1. Add the parent field of the multiple in question (at the DD level of the form's primary file) to your form.
2. Add a pop-up page, dedicated to the DD number of the multiple.
3. Link the pop-up page back to the multiple's parent field through the page's PARENT FIELD property.
4. Add a block to the pop-up page.
5. Add the desired fields from the multiple to the pop-up page's block.

In this lesson, we'll edit the SCHOOL ATTENDED multiple in a pop-up page. First we'll add the SCHOOL ATTENDED field to the appropriate block on an existing page. Then we'll link that field to a pop-up page we'll build, where we will place that multiple's fields.

► To add the SCHOOL ATTENDED multiple field to the Zxxx MAIN 3 block:

1. In the Form Editor, go to the second page of the form (press <PF1><P>, and enter 2).
2. Position the cursor at row 10, column 7, and press <PF2><F> to add a field.
3. Add the field to the block **Zxxx MAIN 3**. Accept the default field order, and make sure the field type is Data Dictionary.
4. In the field properties window, set FIELD to **SCHOOL ATTENDED**. Press <PF1><E> to save the field properties. You should now see a field for SCHOOL ATTENDED on the Form Editor screen.

► **To add a pop-up page for the SCHOOL ATTENDED multiple:**

1. Press <PF2><P> to add a new page. One scheme for numbering pop-up pages is to number them as some fraction added to the page number they pop up from. Since this pop-up page will "pop up" from page 2, for a page number, enter **2.5**.
2. In the Page Properties window, set IS THIS A POP UP PAGE? to **YES**.
3. Set Page Coordinate to **5,5**, and set LOWER RIGHT COORDINATE to **15,75**.
4. Fill in PARENT FIELD. The format for PARENT FIELD is field id,block name,page name. Set PARENT FIELD to "**SCHOOL ATTENDED,Zxxx MAIN 3,2**" (which links the pop-up page to the SCHOOL ATTENDED field on block Zxxx MAIN 3, on page 2).

```
Page Properties

PAGE NUMBER: 2.5
PAGE NAME: Page 2.5
HEADER BLOCK:

PAGE COORDINATE: 5,5
IS THIS A POP UP PAGE? YES
LOWER RIGHT COORDINATE: 15,75

NEXT PAGE:
PREVIOUS PAGE:
PARENT FIELD: SCHOOL ATTENDED,Zxxx MAIN 3,2

DESCRIPTION (WP):
PRE ACTION:
POST ACTION:
```

The pop-up page is linked to the SCHOOL ATTENDED field entirely through the PARENT FIELD property. That's all it takes to invoke the pop-up page!

5. Press <PF1><E> to save the new page. The Form Editor should now display a rectangular box on the screen, from row 5, column 5 to row 15, column 75.

► **To add a block to the pop-up page (since every page needs a block):**

1. Position the cursor at row 6, column 6 on the screen (just inside the top left corner of the box). Press **<PF2>** to add a new block.
2. For block name, enter **Zxxx POPUP 1**. For the "Are you adding Zxxx POPUP 1 as a new block on this page?" question, answer **YES**.

Block Properties Stored in FORM File	
BLOCK NAME: Zxxx POPUP 1	BLOCK ORDER: 1
TYPE OF BLOCK: EDIT	OTHER PARAMETERS...
POINTER LINK:	
PRE ACTION:	
POST ACTION:	
Block Properties Stored in BLOCK File	
NAME: Zxxx POPUP 1	DESCRIPTION (WP):
DD NUMBER: 200.074734	DISABLE NAVIGATION:
PRE ACTION:	
POST ACTION:	

3. In the Block Properties window, the Type of Block should be **EDIT**. Set the DD NUMBER to **200.074734** (tying all fields on the block to the SCHOOL ATTENDED multiple).
4. Press **<PF1><E>** to save the new block.

Finally, now that you've created a pop-up page and a block on the pop-up page, you can add the fields for the SCHOOL ATTENDED multiple to the new Zxxx POPUP 1 block. Add the following fields, in the usual manner, to the new block:

- "School Information" (Caption field)
- SCHOOL ATTENDED
- ADDRESS
- ADDRESS 2
- STATE
- ZIP CODE
- DATE COMPLETED
- VERIFICATION

When you are done adding fields, the pop-up page should look like:

```

                                School Information
SCHOOL ATTENDED: _____
      ADDRESS: _____
ADDRESS 2: _____
      STATE: _____
      ZIP CODE: _____
DATE COMPLETED: _____
      VERIFICATION: _____
```

Now run the form. On the second page, when you select a SCHOOL ATTENDED entry and press <RET>, you should be able to access the SCHOOL ATTENDED multiple through a pop-up page!

```

                                                                 Page 2 of 2
Secondary Menu Option      Synonym
-----
                                School Information
SCHOOL ATTENDED: Southstar
      ADDRESS: 1 Green St.
ADDRESS 2: San Mateo
      STATE: CALIFORNIA
      ZIP CODE: 99998
DATE COMPLETED: JUL 20,1995
      VERIFICATION:
Exit      Save      Refresh
Enter a command or '^' followed by a caption to jump to a specific field.
COMMAND: ██████████      Press <PF1>H for help  ██████████
```

Summary

To edit a multiple when the needed fields won't fit on one screen line, use a pop-up page instead. Place the top level of the multiple as a single field on a block, create a pop-up page for the multiple fields, and link the pop-up page back to the single field using the page's PARENT FIELD property.

2-4. Reposition and Resize the Pop-Up Page

The Form Editor provides ways to visually resize and reposition pop-up pages. In this lesson, you'll reposition and resize the pop-up page you created in the previous lesson for the SCHOOL ATTENDED multiple.

► To reposition the pop-up page:

1. Move to page 2 of the form (press <PF1><P>, and choose page 2).
2. Now move to page 2.5 of the form (press <PF1><P>, and choose page 2.5). The pop-up page should now be displayed, superimposed over page 2.
3. You can use the <TAB> key to move the cursor to each screen element on page 2.5. Press <TAB> repeatedly until the cursor is positioned on the top left corner of the pop-up page.
4. Press <SPACE> or <RET> to select the pop-up page. The four corners of the pop-up page should be highlighted.
5. Experiment with moving the cursor up and down, left and right. The corners will position accordingly, this time representing the new position of the window.
6. Press <SPACE> or <RET> again to deselect the pop-up page, at which time it is repositioned to fit where you moved the page corners.

► To resize the pop-up page:

1. With the pop-up page still displayed, press the <TAB> key until the cursor is positioned at the lower right corner of the pop-up page. Selecting a pop-up page when the cursor is at the lower right corner lets you resize the page.
2. Press <SPACE> or <RET> to select the pop-up page. The four corners of the page should be highlighted. Experiment with moving the cursor up and down, left and right. The corners will position accordingly, representing the new size of the window.

3. Press **<SPACE>** or **<RET>** to deselect the pop-up page, at which time it is resized to fit where you moved the page corners.

```
Page 2 of 2  
  
Secondary Menu Option      Synonym  
-----  
  
School Information  
  
SCHOOL ATTENDED: Southstar  
ADDRESS: asdf  
ADDRESS 2: asdf  
STATE: CALIFORNIA  
ZIP CODE: 99998  
DATE COMPLETED: JUL 20,1995  
VERIFICATION:  
  
File: NEW PERSON (#200)      R5,C5  
Form: ZYXU EDIT  
Page: 2.5 (Page 2.5)  
  
<PF1>Q=Quit  <PF1>E=Exit  <PF1>S=Save  <PF1>V=Block Viewer  <PF1>H=Help
```

Summary

You can visually resize and reposition pop-up pages by selecting them, and pressing the arrow keys to resize and reposition the page.

2-5. Add a ScreenMan Computed Field

A ScreenMan computed field exists only on a form; it's not a data dictionary field. A computed field can be useful to display information that is computed "on the fly".

The value of a ScreenMan computed field is determined by an M expression that sets the variable Y. In the M code (in the field's COMPUTED EXPRESSION property), you can:

- Refer to values of other fields on the current block, by using the other field name in braces. For example, {SECONDARY MENU OPTIONS}.
- Use navigational syntax when referring to fields as well. For example, {SECONDARY MENU OPTIONS:MENU TEXT}.

Let's add a computed field to our form. We'll use it to provide a user-friendly description for the list of secondary menu options we added in a previous lesson. The computed field will use navigational syntax to retrieve and display the MENU TEXT field value (from the OPTION file) based on the value of the SECONDARY MENU OPTIONS field.

► To add the computed field:

1. Return to page 2 of the form (press <PF1><P>, and choose page 2.)
2. On page 2 of the form, position the cursor at row 5, column 50. Press <PF2><F> to add a field at the cursor position.
3. Place the new field on block Zxxx REPEATING 1. FIELD ORDER should be 3, and FIELD TYPE should be **COMPUTED**.

- In the Computed Field Properties window, set the COMPUTED EXPRESSION to be:

```
S Y={SECONDARY MENU OPTIONS:MENU TEXT}
```

This expression uses the field SECONDARY MENU OPTIONS, which is on the same block as the computed field. It also uses navigational syntax to retrieve the value of the MENU TEXT field, which is in the OPTION file (and is linked via the SECONDARY MENU OPTIONS pointer field).

Computed Field Properties

```

FIELD ORDER: 3
OTHER PARAMETERS...
UNIQUE NAME:

CAPTION:
EXECUTABLE CAPTION:

COMPUTED EXPRESSION: S Y={SECONDARY MENU OPTIONS:MENU TEXT}
                
```

SUPPRESS COLON AFTER CAPTION?

- Press **<PF1><E>** in the Computed Field Properties window to save the new field. You should see a third underlined field on the Form Editor screen on row 5, with a length of 1.
- Set the size of the new field. To do this, with the cursor positioned at the new field (its length is only 1), press **<PF3>**. The field should now be highlighted. Also, the R,C row/column indicator in the lower right corner of the screen changes to L=1 (now indicating length of the field is 1). With the field highlighted, use the right arrow key to expand the size of the field until its length is **25** (L=25 in indicator). Press **<PF3>** again to "un-highlight" the field.
- Add two more caption-only fields above the repeating block to add headers for the new MENU TEXT field. To do this, position the cursor on row 3, column 50, and press **<PF2><F>** to add a caption-only field to block Zxxx MAIN 3. Make the caption "**Menu Text**". Then, position the cursor on row 4, column 50, and add another caption-only field with the caption **!DUP("-",25)**.

Now run the form. You'll notice that the menu text for each secondary option, retrieved by the computed field from the OPTION file, is displayed!

Secondary Menu Option	Synonym	Menu Text
XU FINDUSER	FUS	Find a user
XU FIRST LINE PRINT	FLP	First Line Routine Print
XUUSERDISP	DUC	Display User Characterist
+DIUSER	FM	VA FileMan

Summary

Computed fields provide a way to display "on the fly" computed information to the user. Computed fields can be totally stand-alone, or can be based on the values of other fields on the same block.

- This is the end of the second section of the hands-on tutorial. Congratulations! You have explored some of the advanced capabilities of ScreenMan forms, including adding pages to a form, editing multiples, and working with computed fields. If you want to try out some more of the advanced features of ScreenMan, continue on to the next (and last) section of this tutorial!

3. Advanced Features for Your Forms

This unit of the tutorial demonstrates some of finishing touches you can use to fine-tune your forms. It shows how to edit fields in another file, how to use branching logic, and how to use form-only fields to read user input. It also shows how to make a Kernel option out of your form. You will:

- Edit fields in another file, using pointer navigation to reach the other file.
- Add branching logic for fields.
- Use ScreenMan Form-only fields for user input.
- Create a Kernel option to make the form accessible to users.

3-1. Edit Fields in Other Files

When you create a form, you tie the form to a particular file. For example, the form we've been building is tied to the NEW PERSON file.

You can edit fields in another file on your form, however. The easiest way is if there is a pointer field from the form's primary file to the file you want to edit fields in. In this case:

1. Your form's primary file already has a forward pointer field to the other file.
2. You need to add a new block (a block can only hold fields from one DD level).
3. Tie the new block, through its POINTER LINK property, to the field that is the forward pointer in the primary file. The pointer link is the expression that describes the relational jump.
4. Tie the new block, through its DD NUMBER property, to the data dictionary number of the other file.
5. To edit fields in the other file, add them to the block that is tied to the DD number of the other file.

There are other types of relational navigation that can establish the link between files also: backward pointers and lookups.

NOTE: For more information on using other types of relational navigation in ScreenMan forms, please see the "Relational Navigation" chapter in the *VA FileMan Advanced User's Manual (V. 22.0)*.

In this lesson we'll add the SERVICE/SECTION pointer field to our form (it points from the NEW PERSON file to the SERVICE/SECTION file). Then, we'll use that forward pointer field to navigate to the SERVICE/SECTION file. This will let our form users edit the entry in the SERVICE/SECTION file being pointed to by the entry they're editing in the NEW PERSON file.

► **To add the SERVICE/SECTION field to the second page of the form:**

1. On the second page of the form, position the cursor at row 14, column 18. Press <PF2><F> to add a new field at the cursor position.
2. When the Form Editor asks for block, make sure you choose Zxxx MAIN 3. Accept the default field order, and make sure the field type is Data Dictionary.

3. Press **<PF1><E>** to save and exit this dialogue.
4. When you reach the field properties window, set the field to **SERVICE/SECTION**.
5. Press **<PF1><E>** to save the new field.

Now we can add a block to our form to edit fields in the **SERVICE/SECTION** file. This allows users to edit several fields in the **SERVICE/SECTION** file for whichever entry they have selected as their current **SERVICE/SECTION**.

► **To add a block to edit fields in the SERVICE/SECTION file:**

1. On the second page of the form, position the cursor at row 12, column 7. Press **<PF2>** to add a new block at the cursor position.
2. Choose **Zxxx POINTER 1** as the new block name. Answer **YES** to the "Are you adding Zxxx POINTER 1 as a new block on this page?" question.
3. In the Block Properties window, the **TYPE OF BLOCK** should be **EDIT**.
4. In the **DD NUMBER** field, enter **49**. This is the DD number of the **SERVICE/SECTION** file. This means that only **SERVICE/SECTION** file fields can be added as data dictionary fields to this block.
5. In the **POINTER LINK** field, enter **SERVICE/SECTION**. This is the navigational syntax needed to read fields in the **SERVICE/SECTION** file from the **NEW PERSON** file.

6. The block properties should look like:

Block Properties Stored in FORM File	
BLOCK NAME: Zxxx POINTER 1	BLOCK ORDER: 3
TYPE OF BLOCK: EDIT	OTHER PARAMETERS...
POINTER LINK: SERVICE/SECTION	
PRE ACTION:	
POST ACTION:	
Block Properties Stored in BLOCK File	
NAME: Zxxx POINTER 1	DESCRIPTION (WP):
DD NUMBER: 49	DISABLE NAVIGATION:
PRE ACTION:	
POST ACTION:	

INSERT

7. Press <PF1><E> to save the new block.

Now that you have created the new block, add fields to the block (Zxxx POINTER 1) in the usual manner. Position them below the SERVICE/SECTION field. Add the following three data dictionary fields from the SERVICE/SECTION file to the new block (Zxxx POINTER 1):

- MAIL SYMBOL (use a caption of SERVICE MAIL SYMBOL).
- CHIEF (use a caption of SERVICE CHIEF).
- ASST CHIEF (use a caption of SERVICE ASST CHIEF).

To prevent the user from deleting or changing the name of the SERVICE/SECTION entry being edited, we're not letting them edit the .01 field.

Also add caption fields above the four SERVICE/SECTION fields, with the heading "Service Information".

3-2. Add Branching Logic for Fields

You can automatically branch the user from one field on the form to another by setting the variable DDSBR in the BRANCHING LOGIC field property. The BRANCHING LOGIC is executed when the user presses <RET> at the field.

The following variables are defined for any code executed from a field's BRANCHING LOGIC property (and other properties that accept M code):

X	The current internal value of the field.
DDSEXT	The current external value of the field.
DDSOLD	The previous internal value of the field.

To jump to another field on the form, you can set the DDSBR variable in the original field's BRANCHING LOGIC field. You should set the DDSBR variable in the format "Field ID^Block ID^Page ID", where:

Field ID	=	Caption of field, Unique Name of field, or Field Order number of field to jump to.
Block ID	=	Block Order number or Block Name to jump to.
Page ID	=	Page Number or Page Name to jump to.

Let's add branching logic to the OFFICE PHONE field, which is on page 1 of our form. We'll set up the branching logic so that if there is a defined value for OFFICE PHONE, the user will go directly to the SERVICE/SECTION fields on page 2 of the form. If there is not a defined value, the user will stay on page 1.

► To add branching logic to the OFFICE PHONE field:

1. Go to page 1 of the form (press <PF1><P>, and choose page 1).
2. Move the cursor to the OFFICE PHONE field caption, and select it.
3. With the OFFICE PHONE selected, press <PF4> to edit its field properties. In the BRANCHING LOGIC field property, enter:

```
S:$L(X) DDSBR="SERVICE/SECTION^Zxxx MAIN 3^PAGE 2"
```

This code conditionally branches to the field with the "SERVICE/SECTION" caption, on block Zxxx MAIN 3, on Page 2 of the form, provided that there is a defined value for the OFFICE PHONE field.

4. Press <PF1><E> to save the changed field properties.
5. Press <PF1><E> to save the form and exit.

Now try running the form. Try entering a value for OFFICE PHONE. You should branch immediately to the SERVICE/SECTION fields on page 2 of the form. Now go back and delete the value for OFFICE PHONE. In this case, the branching logic should leave you where you are, on page 1 of the form.

Hot Tip



Other Properties Where You Can Set DDSBR

You can also set the DDSBR variable at the field level properties PRE ACTION, POST ACTION, and POST ACTION ON CHANGE, as well as at the form level property DATA VALIDATION. You can set the DDSSTACK variable only at the field level BRANCHING LOGIC property, however. Both of these variables are described in "Appendix A" in this manual and the *VA FileMan Programmer Manual* (V. 22.0).

Hot Tip



Using DDSSTACK to "Stack" Another Form Location

Rather than simply "going" to another location on the form using the DDSBR variable, you can "stack" another location of the form using the DDSSTACK variable. The difference with the DDSSTACK variable is that when a user jumps to another location, does their work, and closes the jumped to page, they are returned to the original page. DDSBR, on the other hand, acts similarly to the M GOTO command; the user is not returned to their original location after a jump. Both of these variables are described in "Appendix A" in this manual and the *VA FileMan Programmer Manual* (V. 22.0).

Summary

You can use branching logic to conditionally branch from one location on a form to another location on the form—even if the locations are on different pages.

3-3. Use Form-Only Fields for User Input

ScreenMan provides Form-Only fields as a way to read user input without tying that input to a dictionary field. They act similarly to the VA FileMan ^DIR reader call, allowing you to read data as a VA FileMan data type (Free Text, Yes/No, Numeric, etc.)

To use form only fields:

1. Create a field and set the field type to FORM ONLY.
2. In the field's properties, under FORM ONLY FIELD PARAMETERS, set the field's READ TYPE property to reflect the type of read you want to do.
3. To retrieve the value of the field in M code, you can refer to the regular variables for a field: X (current internal value), DDSEXT (current external value), and DDSOLD (previous internal value).

Let's add a Form-Only field to the first page of our form, asking the user whether they want to jump to page 2 of the form or not. If they answer yes, we'll use the DDSBR variable again, to branch them to Page 2 of the form.

► **To add a form-only field:**

1. Go to page 1 of the form (press <PF1><P>, and choose page 1).
2. Move the cursor to row 16, column 8. Press <PF2><F> to add a field. Add the field to the Zxxx MAIN 2 block. Choose **Form Only** as the field type.
3. When you reach the field properties screen for the field, set the CAPTION property to **"Go to Page 2"**.
4. Under FORM ONLY FIELD PARAMETERS, set READ TYPE to **"YES OR NO"**. This makes the field read a YES or NO response from the user.
5. Add branching logic in the BRANCHING LOGIC property as follows:

```
S:+X DDSBR="^^Page 2"
```

This branches to page 2, if X (the value of the form-only field in internal format) is positive (if the user answered YES at the field).

6. Press <PF1><E> to save the new field properties.

Exit the Form Editor and try running the form. If you answer yes in the new form-only field, you should jump to page 2 of the form. If you answer no, you should stay on the current page.

Summary

You can use form-only fields to read a variety of input from the user, without that input having to be tied to a data dictionary field.

3-4. Turn Your ScreenMan Form into an Option

Until now, you've been running your form using ScreenMan's Run a Form option. However, end users would need your form set up as an option. Let's turn our form into an option in the menu system.

The two most typical ways to set up an option for a ScreenMan form are:

- Create a Run Routine-type option, and a routine to run
- Create a ScreenMan-type option

In either case (Run Routine or ScreenMan-type option), your option needs to:

1. Determine the record number of the NEW PERSON file entry to edit.
2. Call ScreenMan and load the form once the record number is known (^DDS entry point).

► To create a Run Routine-type option:

Create a routine that determines the record number to edit (use the ^DIC entry point). Then invoke ScreenMan (^DDS entry point) to edit the record. Both of these entry points are described in the *VA FileMan Programmer Manual (V. 22.0)*.

For input variables to ^DDS, you need:

- | | |
|---------|--|
| DDSFIL | The number or global root of the Primary File of the form. |
| DR | The name of the form, enclosed in square brackets. |
| DA | The record number of the file entry to display or edit. |
| DDSPAGE | [optional] The page number of the first page to display to the user. If DDSPAGE is not defined, there must be a page number = 1 in the form. |
| DDSPARM | [optional] Flags controlling return values from ScreenMan. |

The following routine determines the record number of the NEW PERSON file entry to edit, and then calls ^DDS. Clone this routine in your namespace as follows:

```
ZYXWED1 ;isc-sf/fm routine for ScreenMan Tutorial ;5-June-96
        ;;1.0;;
EN
;
N DIC
S DIC="^VA(200," ,DIC(0)="AEMQ" D ^DIC
D:Y>0
.N DIC,DIE,DR,DA,DDSPARM,DDSFILE,DDSPAGE
.S DA=+Y,DDSFILE=200,DR="[ZYXW EDIT]" ,DDSPAGE=1
.D ^DDS
```

Now create a "Run Routine"-type option that calls EN^ZxxxED1 as the routine to run. Run the option. You should now have a Menu Manager option that runs your form!

► **To create a ScreenMan-type option:**

You can quickly set up a ScreenMan-type option to run your form. This option does exactly what the run routine-type option shown above does:

1. Use ^DIC to determine a record to edit (controlled by the DIC {DIC} and DIC(0) fields)
2. Use ^DDS to load your ScreenMan form (controlled by the DR{DDS}, DDSFILE, and DDSPAGE fields)

Set up the option as shown below (substitute your namespace for Zxxx). Then run the option to try it out. Setting up the option this way is quicker than writing a routine!

```
NAME: Zxxx NEW PERSON EDIT           MENU TEXT: Zxxx NEW PERSON EDIT
TYPE: ScreenMan
DIC {DIC}: VA(200,                   DIC(0): AEMQ
DR{DDS}: [Zxxx EDIT]                 DDSFILE: 200
DDSPAGE: 1
UPPERCASE MENU TEXT: Zxxx NEW PERSON EDIT
```

Hot Tip**Record Selection Within ScreenMan**

The options in this lesson determine what record to edit before invoking ScreenMan. To edit a subsequent record, the user must exit ScreenMan, choose another record, and then enter ScreenMan again. You can, however, set up a Record Selection Page to allow record selection to happen within ScreenMan. To do this:

1. Add a pop-up page to your form. Add an edit block to the page.
2. Add one field to this page: a pointer-type form-only field.
3. In the Qualifiers property of the form-only field, specify the number of the file that is your ScreenMan form's primary file.
4. Set the field's caption to something like "Choose a record to edit", and set its data length property to an appropriate size for entering the name of a record in your file.
5. Set the form-level property Record Selection Page to the page number of this page.

Once you've done this, you can invoke ^DDS without specifying the DA of the record to edit; with no DA, ScreenMan will use the record selection page to prompt the user to choose a record. Once they are editing a record, the user can press <PF1>L to choose another record to edit, without leaving ScreenMan.

Summary

To make an option for your form, create a routine that 1) determines the record number to edit from your form's primary file, and 2) calls ^DDS to invoke ScreenMan.

3.5. Where to Go From Here

This is the end of the hands-on tutorial. Congratulations! You've built a ScreenMan form from scratch, enhanced the form with a number of advanced ScreenMan features, and made a Kernel option out of the form. You may want to try creating ScreenMan forms for your own projects now. Feel free to turn back to this tutorial as you do so! Also, turn to the "ScreenMan" chapter in the *VA FileMan Getting Started Manual* (V. 22.0) for a full reference for the ScreenMan Form Editor and other ScreenMan features and the *VA FileMan Programmer Manual* (V. 22.0) for a full reference on the ScreenMan API.

Challenge yourself—they sky is the limit!

The following is just one example of the kind of visually appealing, usable forms that you can create with ScreenMan:

GENERAL INFORMATION (1 of 5)		SUPERMAN, CLARK (47)								
SS No.:	577-88-9898	ADDITIONAL TEST	RESULTS							
Admission:	NOV 29, 1985	+SHIPLEY I.O.	23							
Interview:	JUN 6, 1996	BECK TOTAL SCORE	44							
Time Begun:	08:00A	SCL-90 TOTAL	99							
Time Ended:	09:00A									
Class:	INTAKE									
Contact Code:	IN PERSON									
Gender:	MALE									
Transcriber:	LALIBERTE, DAVID C.									
Interviewer:	LALIBERTE, DAVID C.									
Special:	INTERVIEW COMPLETED									
		SEVERITY PROFILE								
PROBLEMS	0	1	2	3	4	5	6	7	8	9
MEDICAL	**	**	**	**	**	**				
EMP/SUP	**	**	**	**	**	**	**	**	**	
ALCOHOL	**	**	**	**	**	**	**	**		
DRUGS	**									
LEGAL	**	**	**							
FAM/SOC	**	**	**							
PSYCH	**	**	**	**	**	**	**			

Date of the interview. Automatically entered when new.
You may change the date if you wish.

Expect a pause when SAVING data.

Press <PF1>E to Exit
Press <PF1>H for help Insert

NOTE: Some additional features of ScreenMan and the Form Editor are not covered by this tutorial. For example, there are additional properties for fields, blocks, pages, and forms that are not described in this tutorial. For more information on these additional properties, and other ScreenMan and Form Editor features, please see the "ScreenMan" chapter in the *VA FileMan Getting Started Manual* (V. 22.0).

3-6. Delete Your Form

As a final step in the tutorial, you may want to delete the form you've been working on. Should you wish to do so, use the Delete a Form option, which is on the same menu (ScreenMan) as the Edit/Create a Form and Run a Form options.

When you choose a form to delete with the Delete a Form option, it lists all blocks on the form, noting whether or not the blocks are used on any other form. None of the blocks you've used in the tutorial should be part of any other form.

The option asks whether to delete all deletable blocks on your form. For the form you've built for the tutorial, you should answer YES. The option then deletes your form and all deletable blocks on your form.

Appendix A: M Code and Variables in ScreenMan Forms

M code in Field Properties

You can enter M code for the following field properties:

- Branching Logic
- Pre Action
- Post Action
- Post Action on Change
- Data Validation

The M code for these properties can rely on the following variables being defined by ScreenMan:

X	The current internal value of the field.
DDSEXT	The current external value of the field.
DDSOLD	The previous internal value of the field.

You can also enter M code in a field's Executable Caption and Executable Default properties, but you can't depend on the X, DDSEXT, or DDSOLD variables.

The DDSBR Variable

ScreenMan allows you to conditionally branch the user to a field. This can be done with M code in the Branching Logic, Pre Action, Post Action, and Post Action on Change properties at the field level, and in the Data Validation property at the Form level. The M code should set the DDSBR variable to the location of the field to which you wish to take the user. The DDSBR variable should be set in the format "Field ID^Block ID^Page ID", where

Field ID = Field Order number, Caption of field, or Unique Name of field

Block ID = Block Order number or Block Name.

Page ID = Page Number or Page Name

For example:

```
S:X="Y" DDSBR="FIELD 1^BLOCK 1^PAGE 2"
```

This takes the user to the field with unique name or caption "FIELD 1" on the block named "BLOCK 1" on the page named "PAGE 2", if the internal value of the field = Y.

The DDSSTACK Variable

The DDSSTACK variable can be set only in the Branching Logic property of a field. It can be used to branch users to another page when they press the Enter/Return key (<RET>) at the field. After the user closes the page defined in DDSSTACK, ScreenMan takes the user back to the parent page, to the field immediately following the field from which the branch occurred.

Set DDSSTACK equal to the Page Number or Page Name. For example:

```
S:X="Y" DDSSTACK="Page 1.1"
```

This takes the user to Page 1.1 if the internal value of the field is "Y" and the user presses <RET> at the field. When Page 1.1 is closed, the user returns to the parent page, to the field immediately following the field from which the branch occurred.

ScreenMan provides another way to achieve this kind of "branch and return" behavior. You can link a field to a subpage by defining a Subpage Link for the field or by defining a Parent Field for the subpage. The Subpage Link and Parent Field methods, however, do not allow branching conditionally.

Validating Data

ScreenMan uses the definition of a field to automatically validate values inputted by the user. You can use the Data Validation property of a field to validate the value even further. Data Validation is M code that is executed after a user enters a new value for a field and after the automatic validation that ScreenMan normally performs. If the M code sets the variable DDSERROR, ScreenMan rejects the value. You might also want to ring the bell and make a call to HLP^DDSUTL to display a message to the user that indicates the reason the value was rejected.

Other Places for M Code in Forms

Besides the field level, you can use M code at the form, page, and block level. The M code you can use is in property fields at the following levels in a form:

- Form:** Pre Action, Post Action, Data Validation, or Post Save
- Page:** Pre Action or Post Action
- Block:** Pre Action or Post Action (stored in Form File only).
Pre Action or Post Action (stored in Block file).

Appendix B: Tutorial Keystroke Summary

Form Editor—Invoking

D ^DDGF (or use VA FileMan option Edit/Create a Form)

Page Navigation

<PF1><P> Go to Page (Ask which page).
<PF1><PF1><ARROW-UP> Go to Next Page.
<PF1><PF1><ARROW-DOWN> Go to Previous Page.

Screen Element Navigation

<TAB> Move to next screen element.
<Q> Move to previous screen element.

Creating New Elements

<PF2><P> Add a New Page.
<PF2> Add a New Block.
<PF2><F> Add a New Field.

Selecting a Screen Element

1. Position the cursor within the element (field caption, field data, or block).
2. Press <SPACE> or <RET> to select the element.
3. Press <SPACE> or <RET> to de-select the element.

Switching to Block Viewer Screen

<PF1><V> Toggles between Block Viewer screen and normal screen.

Editing Screen Element Properties

- Field: Select the field and press <**PF4**>.
- Field Caption: Position cursor at field and press <**PF3**>.
- Field Data Length: Position cursor at field and press <**PF3**>; use arrow keys to adjust length.
- Block: Switch to Block Viewer Screen (<**PF1**><**V**>), select block, and press <**PF4**>.
- Current Page: Press <**PF4**><**P**>.
- Current Form: Press <**PF4**><**M**>.

Deleting a Page, Block, or Field

Edit the item's properties, and enter an at-sign ("@") in the first field.

Index

A

Adding blocks, 11–13
Adding pages, 37–41

B

Block Viewer screen, 81
Blocks
 Adding, 11–13
 Block Viewer screen, 81
 Deleting, 82
 Display blocks, 19–21
 Edit blocks, 25–26, 27
 Properties, 12, 13, 16, 26, 73, 79
 Repeating blocks, 43–48
Branching logic, 65–66, 77, 78

C

Caption (data dictionary field), 21
Caption prompt shortcuts, 47
Caption-only fields, 15–16
Computed fields, 55–57
Creating forms, 7–9
Cursor
 Position indicator, 12
 Positioning shortcuts, 81

D

Data dictionary fields, 19–21
 Caption, 21
 Data field, 21
DATA LENGTH field property, 29
Data Validation, 78
^DDGF (invoke Form Editor), 8, 81
DDSEXT variable, 65, 66, 77
DDSEXT variable, 65, 77
DDSOLD variable, 65, 77
DDSSTACK variable, 66, 78
Deleting

Blocks, 82
Fields, 82
forms, 73–75
Pages, 82

Display block, 19–21

E

Edit blocks, 25–26, 27
Edit/Create a Form option, 8, 81

F

Fields
 Branching logic, 65–66
 Caption-only, 15–16
 Computed, 55–57
 Data Dictionary, 19–21
 Deleting, 82
 Display Width vs. Field Width, 29
 Display-only, 19
 Form-only, 67–68
 In other files (navigating to), 61–64
 Pointer, 61–64
 Properties, 66, 71, 73, 77, 78
 Repositioning, 15–16
 Selecting, 17

File Manager Access Code, 2

Files (navigating to), 61–64

Form Editor (invoking), 8, 81

Form-only fields, 67–68

Forms

 Deleting, 73–75

 Properties, 66, 71, 73, 79

 Running (Run a Form option), 23

H

Hot Tips

 Block properties (editing), 26

 Caption prompt shortcuts, 47

 Cursor position indicator, 12

- DDSSTACK to "stack" another form location, 66
- Display Width vs. Actual Width for Fields, 29
- Editing block properties, 13, 16
- Moving screen elements, 18
- Page (shortcuts to move to another), 39
- Positioning data dictionary fields, 21
- Properties where DDSBR can be set (branching), 66
- Record selection within ScreenMan, 71

L

- List Form Characteristics option, 31–33

M

- Moving screen elements, 18
- Multiples
 - In a popup page, 49–52
 - In a repeating block, 43–48

N

- Navigating to other files, 61–64

O

- Option (for ScreenMan form), 69–71

P

- Pages
 - Adding, 37–41
 - Deleting, 82
 - Linking, 39

- Moving to another page, 39, 41, 81
- Popup, 49–52
- Properties, 38, 73, 79
- Resizing/repositioning popup pages, 53–54
- Pointer fields, 61–64
- Popup pages, 49–52
 - Resizing/repositioning, 53–54
- PRINT^DDS, 31–33
- Properties
 - Block properties, 12, 13, 16, 26, 73, 79
 - Editing (Keystrokes), 82
 - Field properties, 66, 71, 73, 77, 78
 - Form properties, 66, 71, 73, 79
 - Page properties, 38, 73, 79

R

- Record selection within ScreenMan, 71
- Repeating blocks, 43–48
- Repositioning fields, 15–16
- Resize/reposition popup pages, 53–54
- Running a form, 23

S

- Screen elements
 - Moving, 18
 - Selecting, 81
- Selecting fields, 17

V

- Validating Data, 78

X

- X variable, 65, 77
- XUSCREENMAN security key, 2