

# **Using VPLUS An Intro to Forms Design Self-Paced Training Guide**

**HP 3000 MPE/iX Computer Systems**

**Edition 3**



**Manufacturing Part Number: 32209-90004**

**E0886**

U.S.A. August 1986

---

## **Notice**

The information contained in this document is subject to change without notice.

Hewlett-Packard makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability or fitness for a particular purpose. Hewlett-Packard shall not be liable for errors contained herein or for direct, indirect, special, incidental or consequential damages in connection with the furnishing or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

This document contains proprietary information which is protected by copyright. All rights reserved. Reproduction, adaptation, or translation without prior written permission is prohibited, except as allowed under the copyright laws.

---

## **Restricted Rights Legend**

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013. Rights for non-DOD U.S. Government Departments and Agencies are as set forth in FAR 52.227-19 (c) (1,2).

---

## **Acknowledgments**

UNIX is a registered trademark of The Open Group.

Hewlett-Packard Company  
3000 Hanover Street  
Palo Alto, CA 94304 U.S.A.

© Copyright 1979, 1980, 1986 by Hewlett-Packard Company

## PRINTING HISTORY

New editions are complete revisions of the manual. Update packages, which are issued between editions, contain additional and replacement pages to be merged into the manual by the customer. The dates on the title page change only when a new edition or a new update is published. No information is incorporated into a reprinting unless it appears as a prior update; the edition does not change when an update is incorporated.

The software code printed alongside the date indicates the version level of the software product at the time the manual or update was issued. Many product updates and fixes do not require manual changes and, conversely, manual corrections may be done without accompanying product changes. Therefore, do not expect a one-to-one correspondence between product updates and manual updates.

First Edition	Aug 1979	32209A.00.00
Second Edition	Jan 1980	32209A.02.00
Third Edition	Aug 1986	32209B.04.17

### LIST OF EFFECTIVE PAGES

The list of Effective Pages gives the date of the current edition and of any pages changed in updates to that edition. Within the manual, any page changed since the last edition is indicated by printing the date the changes were made on the bottom of the page. Changes are marked with a vertical bar in the margin. If an update is incorporated when an edition is reprinted, these bars are removed but the dates remain. No information is incorporated into a reprinting unless it appears as a prior update.

**Effective Pages**

**Date**

-----  
all

August 1986

# PREFACE

The *Using VPLUS/V: An Introduction to Forms Design* self-paced training guide is intended to show you how to design forms. The Guide is useful on several levels. Its primary use is as a self-teaching tool to designing forms with FORMSPEC and then entering data in these forms through ENTRY. It can also be used as a guide to using particular features of either FORMSPEC or ENTRY.

You should be familiar with the HP 3000 in order to use this guide. If not, refer to the *HP 3000 Guide for the New User* (32033-90009).

As a training tool, the Guide is most effective if you sit down at a terminal connected to an HP 3000 system with VPLUS/V. Then, following the instructions in the Guide, design an increasingly complex set of forms and enter data into these forms at each step along the way. For best results, at least the first time, you should use the same forms and data illustrated in the Guide.

The Guide starts with a very simple form with no edits and no special processing. It then proceeds step-by-step through the stages of forms design until you have a set of forms that use the highest level of processing. The final forms include automatic calculation of values, passing values between forms, and the use of conditions and phases. Possible relations of one form to another are illustrated through frozen, repeating, and appended forms.

As a guide to specific features of FORMSPEC and ENTRY, you can treat the Guide as a set of separate examples. Each section illustrates a new feature of forms design by showing how this feature is actually used. Before going on to the next feature, it shows how each new form appears to the ENTRY user.

In order to help you find the particular aspect of forms design or data entry that interests you, the first page of each section lists the subjects that are covered in that section. If you come across terms you do not understand, consult the Glossary of Terms at the end of the Guide.

This Guide is not intended to be an exhaustive description of VPLUS/V. For all its features, you should refer to the *VPLUS/V Reference Manual* (32209-90001).

## CONVENTIONS USED IN THIS MANUAL

NOTATION	DESCRIPTION
nonitalics	Words in syntax statements which are not in italics must be entered exactly as shown. Punctuation characters other than brackets, braces and ellipses must also be entered exactly as shown. For example:  EXIT;
<i>italics</i>	Words in syntax statements which are in italics denote a parameter which must be replaced by a user-supplied variable. For example:  CLOSE <i>filename</i>
[ ]	An element inside brackets in a syntax statement is optional. Several elements stacked inside brackets means the user may select any one or none of these elements. For example:  [A] User may select A or B or neither. [B]
{ }	When several elements are stacked within braces in a syntax statement, the user must select one of those elements. For example:  {A} {B} User must select A or B or C. {C}
...	A horizontal ellipsis in a syntax statement indicates that a previous element may be repeated. For example:  [, <i>itemname</i> ]...;  In addition, vertical and horizontal ellipses may be used in examples to indicate that portions of the example have been omitted.
Å	When necessary for clarity, the symbol ~ may be used in a syntax statement to indicate a required blank or an exact number of blanks. For example:  SET[( <i>modifier</i> )]Å( <i>variable</i> );
<u>underlining</u>	When necessary for clarity in an example, user input may be underlined. For example:  NEW NAME? <u>ALPHA</u>

Brackets, braces or ellipses appearing in syntax or format statements which must be entered as shown will be underlined. For example:

```
LET var [subscript] = value
```

Output and input/output parameters are underlined. A notation in the description of each parameter distinguishes input/output from output parameters. For example:

```
CREATE (parm1,parm2,flags,error )
```

shading

Shading represents inverse video on the terminal's screen. In addition, it is used to emphasize key portions of an example.

The symbol `&` may be used to indicate a key on the terminal's keyboard. For example, RETURN indicates the carriage return key.

CONTROLchar

Control characters are indicated by CONTROL followed by the character. For example, CONTROLY means the user presses the control key and the character Y simultaneously.

# Chapter 1            Getting Started with VPLUS

VPLUS/V provides two programs that enable you to design a set of forms to be displayed at a terminal, and then enter data into these forms. The entered data is transferred to a disc file, known as the *batch file*. The two programs are:

- FORMSPEC**            *To design forms.*    The design includes how the form appears on the screen, the sequence in which forms are issued, and rules that determine the type of data to enter and how that data is to be processed.
- ENTRY**                *To enter data.*      Data entry is controlled by the forms design. Forms are issued automatically, data is edited and formatted according to the rules included in the forms design. All acceptable data is written to a batch file. Entered data can subsequently be viewed, printed, or modified.

This section introduces FORMSPEC and ENTRY. Refer to Figure 1-1 for a diagram that shows the relationship between these two programs. The features of each program that are illustrated in this section are:

## Using FORMSPEC

- \* How to run FORMSPEC
- \* Use ENTER to enter data in block mode
- \* Function keys for FORMSPEC
- \* The FORMSPEC menu sequence
- \* How to terminate FORMSPEC

## Using ENTRY

- \* How to run ENTRY
- \* Use ENTER to enter data in block mode
- \* Collect data
- \* Browse and modify data
- \* Function keys for ENTRY
- \* How to terminate ENTRY

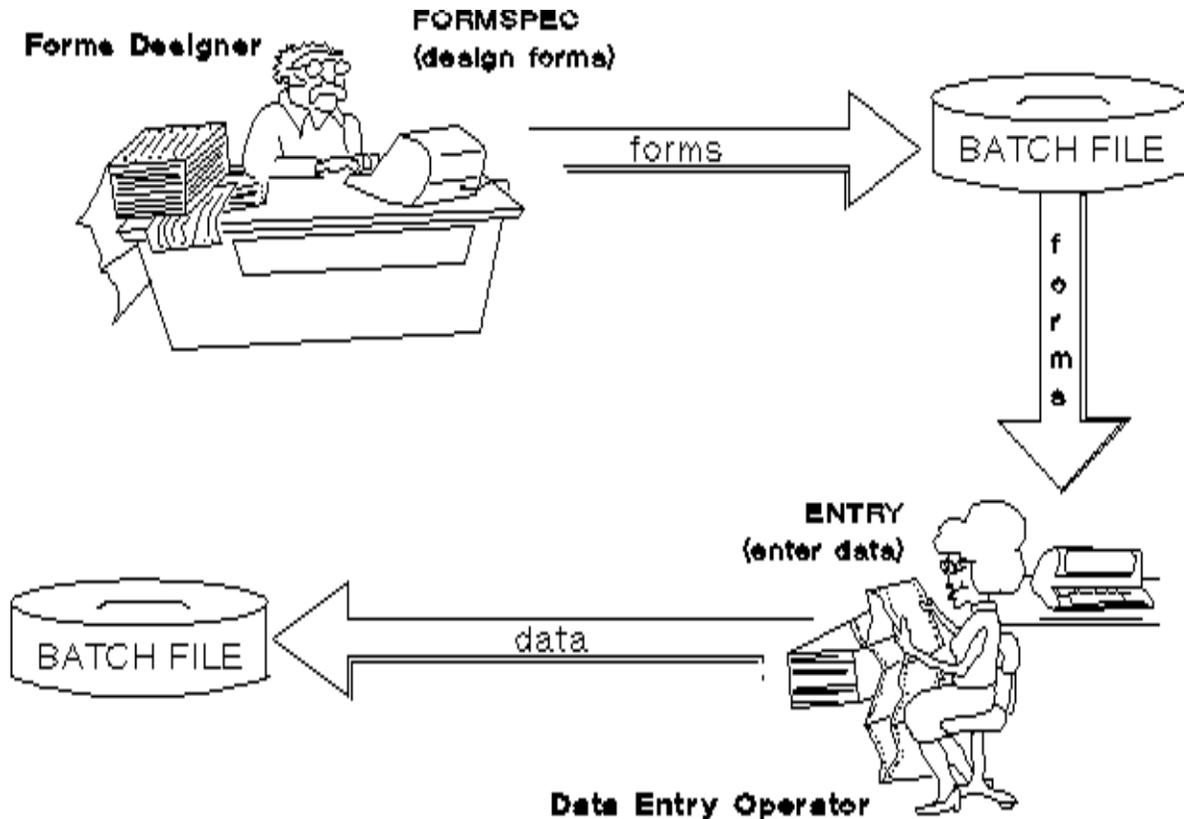


Figure 1-1. Overview of Forms Design and Data Entry

## INTRODUCING FORMSPEC

You must first log on to the HP 3000. Then, in order to design forms with VPLUS/V, you must run FORMSPEC. You enter the following command in response to the MPE colon prompt:

```
:RUN FORMSPEC.PUB.SYS RETURN
```

When this command is accepted, FORMSPEC is in control. You use FORMSPEC to perform the following functions:

- \* Create and name a forms file to contain your forms.
- \* Name each form and specify the sequence in which it is displayed.
- \* Design the form layout for each form.
- \* Specify the attributes for each field in a form, unless you use the default field attributes supplied by FORMSPEC.
- \* Compile the forms file so that it can be used by an application.

## Block Mode

FORMSPEC operates in block mode. In block mode, all data typed in unprotected areas on the screen is sent to FORMSPEC at one time when you press ENTER, rather than line-by-line as with RETURN. FORMSPEC issues "menu" screens on which you type any necessary information into "unprotected" areas of the screen. (Unprotected areas are highlighted.) You move to the start of each successive field by pressing TAB. The next sections, in which you design increasingly complex forms, describe in detail the information you enter on the FORMSPEC menus.

## FORMSPEC Function Keys

Before running FORMSPEC, you should understand how your terminal function keys are used for FORMSPEC. The functions executed when you press these keys correspond to the labels displayed on the terminal screen by FORMSPEC, as shown in Figure 1-2.

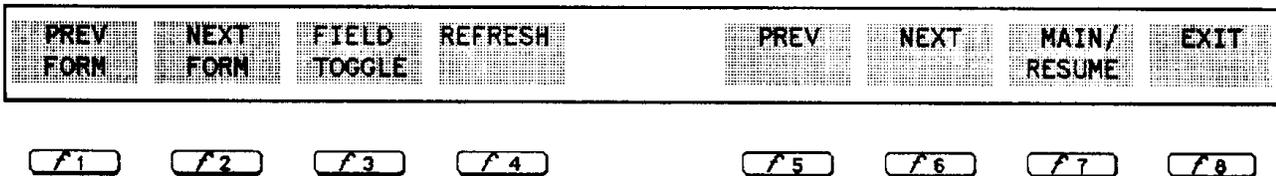


Figure 1-2. Function Key Labels for FORMSPEC

The FORMSPEC labels and their functions are:

PREV FORM	Press this key to display the previous Form Menu.
NEXT FORM	Press this key to display the next Form Menu.
FIELD TOGGLE	Press this key to enter processing specifications on the Field Menu.
REFRESH	Press this key to redisplay the current menu in its initial state.
PREV	Press this key to display the previous menu.
NEXT	Press this key to display the next menu.
MAIN/RESUME	Press this key to request the Main Menu or to return to current menu.
EXIT	Press this key to exit from FORMSPEC.

The use of each function key is described in more detail in later sections.

## **Menu Sequence**

The menus issued by FORMSPEC have a basic sequence that can, however, be varied through use of the function keys PREV FORM or NEXT FORM, PREV or NEXT, and the MAIN/RESUME key. In addition, the Main Menu provides great flexibility in selecting what you want to do, whether it is adding a form, modifying an existing form, or changing the characteristics of all the forms in the file, called *globals*. Figure 1-3 illustrates the basic FORMSPEC menu sequence.

## **Terminate FORMSPEC**

You terminate FORMSPEC and return to MPE control by pressing the function key labeled EXIT, which is f8 . The next time you run FORMSPEC and ask for an existing forms file, you continue forms design wherever you leftoff.

Note that you should compile the forms file before exiting from FORMSPEC. This insures that any additions or changes to the forms design are translated into machine-readable form so that the file can be used by a data entry program. A compiled forms file can still be modified since the form definitions are kept in two versions: a "source" version for easy modification and a "compiled" version for data entry.

When you run FORMSPEC again after terminating, the Main Menu is issued so that you can select what you want to do. For example, if you want to view and/or change an existing Field Menu, you select the menu by the name of the field and the form in which it appears. FORMSPEC then displays the selected menu. Or, if you want to continue adding forms, simply request "Add a Form" and a blank Form Menu is displayed. Forms are kept in alphabetical order, but this does not necessarily determine the order in which forms are executed. The sequence in which forms are executed is part of the forms design itself and can be varied easily.

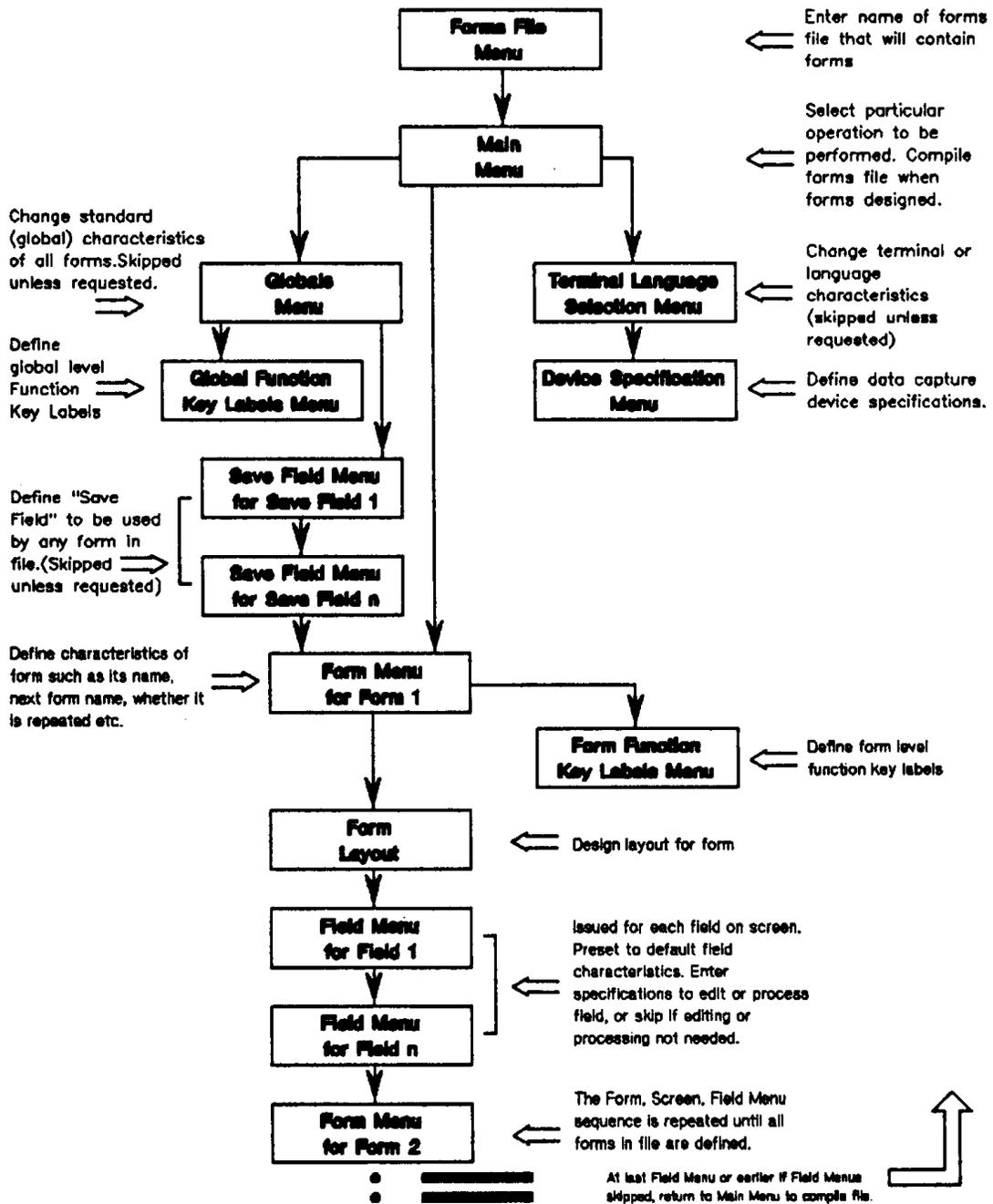


Figure 1-3. Menu Sequence in FORMSPEC

### INTRODUCING ENTRY

In order to enter data into a set of forms, you run

ENTRY. The forms must be in a forms file designed through FORMSPEC. To run ENTRY, enter the following command in response to the MPE colon prompt:

:RUN ENTRY.PUB.SYS RETURN

When the operating system accepts this command, ENTRY is in control. You use ENTRY to perform the following data entry functions:

- \* Name the forms file containing the forms to be displayed.
- \* Create a batch file to contain the entered data or access an existing batch file.
- \* Enter data on the displayed forms, correct any errors detected by ENTRY. Valid data is written to records in the batch file.
- \* Look at the data written to the batch file records.

After you name the forms file and batch file in line mode, ENTRY transfers to block mode, described earlier.

### **Data Collection**

The data is typed into the "unprotected fields" on the forms displayed at the terminal by ENTRY. When you have typed in all the data in a form, press ENTER. ENTRY then processes the data according to the specifications for the form defined through FORMSPEC. If it detects any errors, ENTRY re-displays the form, highlights the fields with errors, and displays an error message for the first error. When all errors have been corrected, ENTRY accepts the data, writes it to the named batch file, and displays the next form.

The sequence of the forms is completely determined at design time. Your only control over forms sequence in Collect mode is through function keys that allow you to request the "head" form or to stop a repeating form.

### **Browse and Modify**

The data entered through each form is written to a single record in a batch file maintained by ENTRY. You may look at and correct, if desired, the data in each record. ENTRY has a browse/modify mode selected by pressing a function key. In this mode, you are shown the data previously entered and edited in any form. Using function keys you can display any previously entered data. The data is shown as it appeared when you entered it, and you can correct data at this point and then press ENTER to record these corrections in the batch file.

The next sections of this manual illustrate, through working examples, exactly how you enter data, view the entered data, and modify it if needed.

### **ENTRY Function Keys**

The function keys on the terminal keyboard are assigned special func-

tions for ENTRY. These functions differ slightly depending on whether you are operating in browse/modify or data collection mode. The function keys are illustrated in the following diagrams, Figure 1-4 for data collection and Figure 1-5 for browse/modify mode.

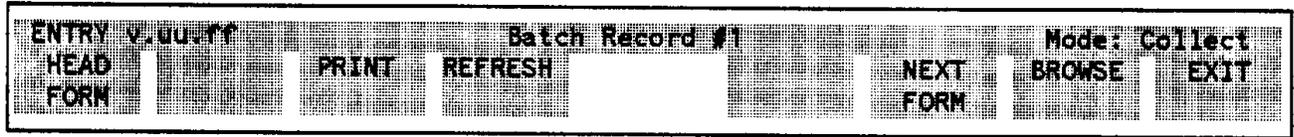


Figure 1-4. Function Key Labels for ENTRY in Collect Mode

The ENTRY labels and their functions are:

- HEAD FORM            Press to display the first form (as defined in FORMSPEC).
- PRINT                Press to print current form on line printer.
- REFRESH             Press to clear data from current form.
- NEXT FORM           Press to terminate display of repeating form and display next different form.
- BROWSE              Press to enter Browse/Modify mode.
- EXIT                 Press to exit from ENTRY.

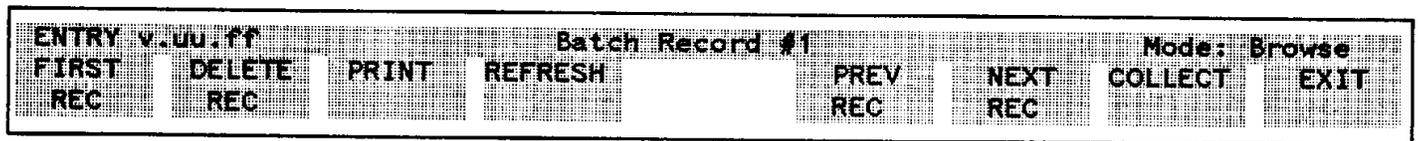


Figure 1-5. Function Key Labels for ENTRY in Browse Mode

The ENTRY labels and their functions are:

- FIRST REC            Press to display the first record in data file.
- DELETE REC          Press to delete current record.
- PRINT                Press to print current record on line printer.

REFRESH            Press to clear corrections from current record.

PREV REC           Press to display previous record.

NEXT REC           Press to display the next record. If the record already displayed is the last record, a message is displayed that states there are no more batch records.

COLLECT            Press to return to Data Collection mode.

EXIT                Press to exit from ENTRY.

**Terminating ENTRY**

You can terminate ENTRY and return to MPE control at any time by pressing EXIT. Any data entered with ENTER is recorded and written to the data file. When you run ENTRY again with the same batch file, the next form in sequence is displayed. The next record written to the batch file follows the last record written in a previous session.

## Chapter 2 Design and Use a Simple Form

Section 1 introduced you to FORMSPEC and ENTRY. In this and the following sections, you will use both programs to create a simple data entry application. In the process, you will come to understand and be familiar with the features and capabilities of both FORMSPEC and ENTRY.

Throughout the exercises in this guide, you will see the menu or form under discussion displayed on the left page and the accompanying discussion on the right. To accomplish the exercise, you perform the actions set off in the boxes labeled **Action**. In general, the features and capabilities of the programs are explained in the discussions of the exercise. In some cases, explanations are set off as separate topics that allow you to take a break from the exercise and study the concepts in more detail. At these points, you are encouraged to read the portions of the *VPLUS/V Reference Manual* recommended as part of the topics.

To illustrate the basic simplicity of FORMSPEC, this section shows you how to create a forms file with only one form, no edits, and no special processing. Then, using ENTRY, you enter data into this form.

### Using FORMSPEC:

- \* Use the Forms File Menu to create the forms file
- \* Use the Main Menu to add a new form
- \* Use the Form Menu to name a new form fields
- \* Design the new form on the Form Layout Menu
- \* Use the Field Menu--defaults only
- \* Use NEXT to display the next menu
- \* Use MAIN/RESUME to return to the Main Menu
- \* Use the Main Menu to compile the forms file
- \* Use EXIT to exit from FORMSPEC

### Using ENTRY

- \* Request the forms file and file
- \* Enter data in displayed forms
  - \* Learn how to tab between
- \* Learn how to correct typing errors
- \* Use EXIT to exit from ENTRY

---

```
| :RUN FORMSPEC.PUB.SYS |
```

---

Figure 2-1. Run FORMSPEC

# Run Formspec

The first step in forms design is to decide on the layout of the form or forms that you need for your application. You may have an established form or you may be starting from scratch. In any case, you generally need a rough draft of the forms to be created as the first step in forms design with FORMSPEC. However, in this guide this first step is taken for you. A simple form is provided in this section; increasingly complex edits and additional forms are provided in later sections. For best results, you should use the same forms and data illustrated in this guide.

## Action

You start this exercise by issuing the command to run FORMSPEC, as shown in Figure 2-1.



Figure 2-2. The Forms File Menu with File Name

## Specify a Forms File Name

Once the RUN command is accepted, FORMSPEC clears the screen, then displays the Forms File Menu shown in Figure 2-2, which is always the first menu displayed by FORMSPEC. Before you can create or modify any forms, you must specify the name of the forms file. The forms file name must be a valid MPE file name, which means it has up to eight characters,

of which the first must be alphabetic. FORMSPEC accepts a fully qualified file name, which can include the group and account in which the file is stored, as well as any lockwords assigned for security.

### **Syntax**

```
filename {/lockword }[.groupname [.acctname ]]
```

With up to eight characters per name, and the three separators (slash and periods) you can see that if you specify the fully qualified file name, it can take up to 36 characters. This is why the field for the forms file name is so long.

### **Action**

Use ORDFORM1 for the name of the file to contain your forms file. Type in the file name, as shown in Figure 2-2, and press ENTER.

Since this is a new forms file, a message is displayed (replacing the menu title) that states:

```
File does not exist. Press ENTER to create file.
```

### **Action**

Press ENTER again as the message directs.

```

FORMSPEC v.uu.ff Main Menu                                FORMS FILE: ORDFORM1
[A] Enter Selection
Forms file is 1% full
A--Add a form
S--Add a Save field
T--Terminal/Language Selection Menu
G--Go to GLOBALS Menu, OR Go to form [ ] field [ ]
L--List Forms File, OR List form ... [ ]
D--Delete Save field ..... [ ]
   Form ..... [ ]
C--Copy new form name ..... [ ]
   from form ..... [ ]
   from Forms File (opt) ..... [ ]
X--Compile Forms File
   Optional: Fast Forms File [ ]
R--Relate child form [ ] to parent form [ ]

PREV FORM  NEXT FORM  REFRESH  PREV  NEXT  MAIN/RESUME  EXIT

```

08:29

Figure 2-3. The Main Menu

## Select the ADD Option

After you type in the file name and press ENTER twice, FORMSPEC creates the forms file. FORMSPEC then displays the next menu in the menu sequence, the Main Menu, shown in Figure 2-3.

### Action

In general, after you create a forms file the first option you select is to add a form. You specify add a form by typing the letter A in the Main Menu selection box, as shown in Figure 2-3. Press ENTER.

```

FORMSPEC v.uu.ff Form Menu                                FORMS FILE: ORDFORM1

Form Name [SHIPTO ]

Repeat Option [N]
N--No Repeat
A--Repeat, appending
R--Repeat, overlaying

Next Form [C]                                           Name [$HEAD ]
C--Clear before Next Form
A--Append Next Form
F--Freeze, then append Next Form

Function Key Label [ ]
Y--Define form level function key labels.

Reproduced from [ ] (opt)

Comments [Only 1 form in this file.]

PREV FORM  NEXT FORM  REFRESH  PREV  NEXT  MAIN/RESUME  EXIT

```

Figure 2-4. The Form Menu for the SHIPTO Form

## Defining the New Form

Once you select the add option at the Main Menu and press ENTER, FORMSPEC displays a Form Menu, as shown in Figure 2-4. On the Form Menu, you specify the characteristics of the form, which are the form name, the form sequencing options, whether the form is reproduced from another form, and a comment, if desired. Also, you can specify on the Form Menu that you want to define form level function key labels, in which case an additional menu (not shown) is displayed on which you specify the labels.

### Action

Specify SHIPTO for the name of the form you are to design.

The Repeat Option, the Next Form and the Name fields contain the default values specified by FORMSPEC, which are N, C, and \$HEAD respectively. Keep the default values for now. These codes mean that the form is displayed once and then the screen is cleared before the next form is displayed.

The Function Key Label and the Reproduced from fields are blank; leave them that way.

To specify a comment, use TAB to tab past the other fields to the

beginning of the comment field and type in the comment. Match the comment shown in Figure 2-4 or make up your own comment.

Press ENTER to record the form characteristics for the SHIPTO form.

## **Topic: Form Menu Fields**

The next step is to actually layout the new form on the blank menu that FORMSPEC displays for this purpose. But before doing so, read the following descriptions of the fields displayed on all Form Menus.

### **The Form Name Field**

You must give each form a unique name of up to 15 characters, with the first character alphabetic.

### **The Repeat Option Field**

The Repeat Option controls whether this form is displayed once or multiple times. If the later, the form can be appended or overlaid.

### **The Next Form and Name Fields**

The next form is, in this case, not given a name. Instead, the term \$HEAD is used, which is the default value for Next Form. It is a special term that means the first form to be executed. Since there is only one form in the forms file, \$HEAD in this case means that the first form is displayed when ENTER is pressed. The effective result is that the form is displayed anew each time ENTER is pressed during execution of ENTRY. (Note that each time data is entered into this form, the data is written to a separate record in the data file.)

### **Action**

For more information on the form sequencing options, complete with examples, refer to "Understanding Form Sequencing" in Section 3 of the *VPLUS/V Reference Manual*.

### **The Function Key Label Field**

The Function Key Label Field is used to define form level function keys. If you type a Y and press ENTER, the Form Function Key Menu appears (not shown in this guide). The default is a blank field ( ) for no form labels.

### **Action**

For more information, refer to the "Form Function Key Label Menu" in Section 3 of the *VPLUS/V Reference Manual*.

## The Reproduced From Field

The Reproduced From field is used for the creation of family forms.

### Action

For more information, refer to "Form Families" in Section 3 of the *VPLUS/V Reference Manual*.

## The Comments Field

In the Comments field, you can enter any comment you please or, if you want to leave it blank, simply press ENTER when the Form Menu is displayed--all default values are assumed in this case. If a comment is specified, as was shown in Figure 2-4, it is solely for the use of the FORMSPEC forms designer. The comment does not appear when ENTRY is run.

Design New Screen and Press ENTER



Figure 2-5. The Form Layout Menu

## Layout a New Form

Once you press ENTER to record the form characteristics, FORMSPEC clears the screen. As shown in Figure 2-5, a message appears briefly in the upper left corner of the screen. When the message disappears, it leaves the screen as a blank pallet, ready for you to layout a new form. To layout the form, you need to remember only two rules:

1. **Delimit all fields.** Any area in which you want data entered is called a field. Use brackets ([]) to indicate the beginning and end of each field. Or, if you do not want the field delimiters to

appear on the form, press ESCAPE before typing each bracket. Combining ESCAPE and [ indicates the start of a field. Similarly, combining ESCAPE and ] indicates the end of a field. When preceded by ESCAPE, the brackets do not appear on the form and do not take up any room; for this reason they are called **nonprinting brackets**. With either printing or nonprinting brackets, you can fill the field with dots (periods) so you can see the extent of the field. The dots only appear during form layout; they are not shown when the form is displayed by ENTRY.

2. **Name each field within the delimiters.** A field name, called the **field tag**, must be typed inside the field delimiters. This tag can be upper or lower case letters or a combination of upper and lower case. The first character must be a letter, but other characters may be letters, the numbers from 0-9, or an underline (\_).

All other considerations of form layout are aesthetic. During form layout, you can use all the terminal edit and display group keys and the cursor control keys (as described in your terminal reference manual). These keys allow you to move freely around the screen inserting or deleting lines and characters and moving the fields around until you are satisfied with how the form appears. Do not enable automatic line feed (Auto LF) while running FORMSPEC.

```

-
                                     *****ABC MANUFACTURING*****
                                     Date: [ordate                ]

SHIP TO:
  Name [name                ]
Address [address            ]
  City [city                ] State [st]
  Zip [zip                  ] Telephone [phone                ]

PREV  NEXT  REFRESH  PREV  NEXT  MAIN/  EXIT
FORM  FORM  FORM      FORM  FORM  RESUME FORM

```

**Figure 2-6. The Form Layout for the ShipTo Form**

### Action

Layout the form as shown in Figure 2-6; the positions of the fields are defined in Table 2-1.

Each field into which the user can enter data has been delimited by

printing brackets. Within these brackets, a field tag names each field. For example, ordate identifies the field into which a date is entered, and name identifies the field into which the customer name is entered. The other field tags are address, city, st, zip, and phone. These tags never appear when the form is displayed. All the other text, which consists of the title and the labels on the fields, will be displayed exactly as it is typed on the screen during form layout.

**Table 2-1. Defining the Fields for the ShipTo Form**

Field	Tag	Line No.	Length
Date	ordate	5	18
Name	name	10	42
Address	address	11	42
City	city	12	30
State	st	12	2
Zip	zip	13	10
Telephone	phone	13	14

**Action**

When the form is laid out to your satisfaction, press ENTER to record it in the forms file.

The next step is to proceed to the Field Menu for the first field on this form.

```

FORMSPEC v.uu.ff Field Menu                                FORM: SHIPTO
*****ABC MANUFACTURING*****

Date: [ordate      ]
      ^
Num [1  ] Len [18  ] Name [ORDATE  ] Enh [HI  ] FType [0] DType [CHAR]
Initial Value [

```

---

```

*** Processing Specifications ***

```

```

PREV  NEXT  FIELD  REFRESH          PREV  NEXT  MAIN/  EXIT
FORM  FORM  TOGGLE                    RESUME

```

Figure 2-7. The Field Menu for the Date Field

## Using the Default Field Edits

Once you press ENTER to record the new form, the next menu displayed by FORMSPEC is the Field Menu for the first field in the form. In this exercise, it is the Field Menu for ORDATE, as shown in Figure 2-7.

FORMSPEC displays a Field Menu for each field identified on the form. These menus are displayed in a sequence, called **screen order**, based on the order the fields appear on the form, counting from left to right, top to bottom. Each Field Menu displays the attributes of that field, its field number, length, name, and so forth. Any of the field attributes, except the field number and the field length, can be changed on the Field Menu. Above the list of attributes is a portion (up to 3 lines) of the form that includes the field being described. A mark (^) indicates the field.

If you make changes to a menu, you must press ENTER. But, if you make no changes and want to simply look at other Field Menus, use the function keys NEXT or PREV to get to the other Field Menus, instead of pressing ENTER.

### Action

Look at the Field Menu for ORDATE, but do not change anything on the menu. For this exercise, leave all the fields of the SHIPTO form with the default field attributes assigned by FORMSPEC. You can look at the Field Menu for each of the other fields by pressing NEXT or PREV to step through the menus.

Since you are using all default values for the fields in the SHIPTO form, the form is now completely defined. Press MAIN/RESUME to go to the Main Menu.

```

FORMSPEC v.uu.ff Main Menu                                COMPILING FORM: SHIPTO
[X] Enter Selection
A--Add a form
S--Add a Save field
T--Terminal/Language Selection Menu
G--Go to GLOBALS Menu, OR Go to form [          ] field [          ]

L--List Forms File, OR List form ... [          ]

D--Delete Save field ..... [          ]
   Form ..... [          ]

C--Copy new form name ..... [          ]
   from form ..... [          ]
   from Forms File (opt) ..... [          ]

X--Compile Forms File
   Optional: Fast Forms File [          ]

R--Relate child form [          ] to parent form [          ]

PREV  NEXT  REFRESH  PREV  NEXT  MAIN/  EXIT
FORM  FORM  FORM      FORM  FORM  RESUME  FORM

```

Figure 2-8. The Main Menu

## Compile the Forms File

Before the SHIPTO form can be used for data entry, the forms file ORDFORM1 must be compiled. The process of compilation changes the form and field definitions into machine-readable form so that the forms file can be used for data entry. When you exit from FORMSPEC, the compiled forms file is kept in your account with the name specified on the Forms File Menu. Note that a compiled forms file maintains two versions of the form and field definitions: one version, the "source" can be modified through FORMSPEC; the other "compiled" version is used by applications, such as ENTRY.

### Action

Type an X in the selection box and press ENTER. The current file is compiled.

While the file is being compiled, the forms file name in the window line at the top of the menu is replaced by the word COMPILING, followed by the name of the particular form being compiled, as shown in Figure 2-8. The COMPILING message blinks on and off during the compilation. When the file is completely compiled, the window line again displays the forms file name.

```

FORMSPEC v.uu.ff Main Menu                                FORM: SHIPTO
[ ] Enter Selection
A--Add a form
S--Add a Save field
T--Terminal/Language Selection Menu
G--Go to GLOBALS Menu, OR Go to form [ ] field [ ]
L--List Forms File, OR List form ... [ ]
D--Delete Save field ..... [ ]
   Form ..... [ ]
C--Copy new form name ..... [ ]
   from form ..... [ ]
   from Forms File (opt) ..... [ ]
X--Compile Forms File
   Optional: Fast Forms File [ ]
R--Relate child form [ ] to parent form [ ]

PREV  NEXT  REFRESH  PREV  NEXT  MAIN/  EXIT
FORM  FORM  FORM      FORM  FORM  RESUME FORM

```

Figure 2-9. The Main Menu

## In Case of Errors

When the file is completely compiled, the window line again displays the forms file name, as shown in Figure 2-9.

If you attempt to exit without compiling the forms file, a message is displayed that states:

WARNING: Forms file modified and not compiled. Press EXIT to exit.

Be sure to compile the forms file before exiting.

If you have made any mistakes in defining a form, a message is displayed in the window line during compilation. The message directs you to press NEXT to see the menu in which the error occurred. If you press NEXT, the menu with the error is displayed. When you correct the error and press ENTER the correction is recorded and the Main Menu is displayed. Again type an X in the selection box and press ENTER to compile the file.

### Action

When the forms file ORDFORM1 is compiled, exit from FORMSPEC by pressing EXIT, which returns you to MPE control. The colon prompt (:) is displayed on the screen. You can then run FORMSPEC again, run any other program on your system, run ENTRY to enter data, or log off.

```
:RUN ENTRY.PUB.SYS
```

```
HP32209v.uu.ff ENTRY (C) HEWLETT-PACKARD CO. 1986
```

```
ENTER FORMS FILE NAME AND PRESS RETURN: ORDFORM1
```

```
ENTER BATCH FILE NAME AND PRESS RETURN: CUSTLIST
```

**Figure 2-10. Run ENTRY and the SHIPTO Form, Ready for Record No. 1**

## Run Entry and Display Form

Once the forms file is compiled, you can enter data into any form in that file through ENTRY. Run ENTRY as shown in Figure 2-10. When the RUN command is accepted, ENTRY displays two prompts, as shown. The first requests the forms file name and the second asks for the name of the batch file. The batch file is the file to which the entered data is written.

### Action

In response to the forms file prompt, type the name ORDFORM1 since this is the name of your compiled forms file. Then in response to the batch file prompt, for this exercise identify the batch file by the name CUSTLIST.

In general, the file names you use with ENTRY can be any valid MPE file name, meaning a name of up to eight characters that starts with a letter of the alphabet. The file name can be qualified by your *groupname* and *acctname*. Thus, if your group is MASTER and your account is ORDERS, the names could be entered as:

```
ORDFORM1.MASTER.ORDERS  
CUSTLIST.MASTER.ORDERS
```

Once you have answered the two prompts, ENTRY opens the forms file ORDFORM1 and creates a new batch file with the name CUSTLIST. Then ENTRY clears the screen and displays the first form in the forms file. In this case, only one form was defined, the SHIPTO form, as shown in Figure 2-10.

You should see the cursor (the blinking underline) indicating the position where you can type in a character. The cursor is positioned to the first unprotected field into which you can enter data. Notice that all the fields are displayed by ENTRY in inverse video, which is the default field enhancement specified by FORMSPEC.

\*\*\*\*\*ABC MANUFACTURING\*\*\*\*\*

Date: [September 22, 1986]

SHIP TO:

Name [John P. Morgan ]  
Address [1100 Commercial Way ]  
City [Los Angeles ] State [CA]  
Zip [90219 ] Telephone [(213) 555-7890]

```
ENTRY v.uu.ff          Batch Record #1          Mode: Collect
HEAD |                | PRINT | REFRESH |                | NEXT | BROWSE | EXIT
FORM |                |     |       |                | FORM |       |
```

Figure 2-11. The SHIPTO Form, Record No. 1 with a Typing Error

## Enter Data

### Action

Type in the date September 22, 1986, as shown in Figure 2-11.

Since the date fills the field, the cursor is positioned to the beginning of the next unprotected field, the Name field. Note that if data does not fill a field, you must press TAB to move the cursor to the next field.

### Action

Continue typing in data until each field contains a value, as shown in Figure 2-11. Press TAB to move to each field.

As currently defined, you can skip fields if you want, since all fields on this form are optional--this is a default attribute of FORMSPEC fields that allow you to leave any field blank.

Before you press ENTER to record the data, always look over the form to see if there are any mistakes. Note that there is one mistake in Figure 2-11; the word September in the date field is misspelled.

\*\*\*\*\*ABC MANUFACTURING\*\*\*\*\*

Date: [September 22, 1986]

SHIP TO:

Name [John P. Morgan ]  
Address [1100 Commercial Way ]  
City [Los Angeles ] State [CA]  
Zip [90219 ] Telephone [(213) 555-7890]

ENTRY v.uu.ff	Batch Record #1	Mode: Collect			
HEAD	PRINT	REFRESH	NEXT	BROWSE	EXIT
FORM			FORM		

Figure 2-12. Corrected SHIPTO Form, Record No. 1

## Correct Typing Errors

Before you can correct a typing error, such as the misspelling shown in Figure 2-12, you must position the cursor at the field to be corrected. If the cursor is above the field, where "above" means closer to the top of the terminal screen than the field to be corrected, then you simply press TAB until the cursor is at the desired field. If the cursor is below the field, where "below" means closer to the bottom of the terminal screen, then you must move the cursor towards the top of the screen. Since the direction is the reverse of that provided by TAB, it is called **backtab**. How you backtab is determined by the type of terminal that you are using:

- \* On HP 264X terminals, you backtab, which is to move from the current field to the one above, by pressing CONTROL and TAB simultaneously.
- \* On HP 262X and HP 239X terminals, and on HP personal computers, you press SHIFT and TAB simultaneously to backtab.

Another method is to press TAB repeatedly until the cursor reaches the desired field.

Once you position the cursor at the field to be corrected, you simply type over the data in the field with the correct data.

**Action**

Practice backtabbing, using the method appropriate to your terminal.

Correct any fields with typing errors. When your data matches Figure 2-12, press ENTER to record the data for Record #1.

The cursor can be anywhere on the form when you press ENTER.

```
-
                                     *****ABC MANUFACTURING*****
                                     Date: [          ]

SHIP TO:
  Name [          ]
  Address [          ]
  City [          ] State [  ]
  Zip [          ] Telephone [          ]

ENTRY v.uu.ff          Batch Record #2          Mode: Collect
HEAD |          | PRINT | REFRESH |          | NEXT | BROWSE | EXIT
FORM |          |      |        |          | FORM |       |

```

**Figure 2-13. The SHIPTO Form, Ready for Record No. 2**

Once you press ENTER to write the data to the batch file, ENTRY then displays the next form in the forms file. Since there is only one form in this forms file, ENTRY clears the screen and displays the SHIPTO form. ENTRY is now ready for Record #2, as shown in Figure 2-13. The current record number is always displayed in the window line.

**Action**

If you specify data on this form and press ENTER, the data specified is written to Record #2 of the batch file CUSTLIST. The SHIPTO form is displayed, ready for Record #3, and so on. You may continue entering data in this manner if you so desire.

Press EXIT to terminate ENTRY.

## Chapter 3 Simple Editing

The ORDFORM1 forms file, as it is currently designed (refer to Section 2), accepts any data entered by the user or allows the user to leave all fields blank. This section shows how to modify the forms file to add simple field type and data type edits that cause standard checks to be performed on entered data.

For the user, this section shows how to correct errors detected by ENTRY and also shows how to browse through the data records entered in the batch file.

### Using FORMSPEC

- \* Locate an existing form (Main Menu)
- \* Change the form name (Form Menu)
- \* Add fields to the form (Form Layout Menu)
- \* Specify simple edits (Field Menu)
  
- \* Change the field name (Field Menu)
- \* Change the field type (Field Menu)
  
- \* Assign an initial value (Field Menu)
  
- \* Change the date type (Field Menu)
- \* Change the numeric type (Field Menu)

### Using ENTRY

- \* Request the forms file and a new batch file
- \* Change the initial value
- \* Enter data in edited fields
- \* Observe the enhancement of fields in error
- \* Correct errors
- \* Use BROWSE to go to Browse mode
- \* Use FIRST REC, NEXT REC, and PREV REC
- \* Review entered data
- \* Use COLLECT to return to Collect mode

---

```
| :RUN FORMSPEC.PUB.SYS |
```

---

Figure 3-1. The Forms File Menu

## Run FORMSPEC

In this exercise, using FORMSPEC, you add fields to the form you created in Section 2. You then define simple edits for several fields in the form.

### Action

Run FORMSPEC as shown in Figure 3-1. Press RETURN to transmit the command to MPE. When FORMSPEC issues the Forms File Menu, specify the name of the forms file (ORDFORM1), as shown in Figure 3-1. Press ENTER to transmit your

file request to FORMSPEC.

```
FORMSPEC v.uu.ff Main Menu                                FORMS FILE: ORDFORM1
[G] Enter Selection
Forms file is 1% full
A--Add a form
S--Add a Save field
T--Terminal/Language Selection Menu
G--Go to GLOBALS Menu, OR Go to form [SHIPTO ] field [ ]
L--List Forms File, OR List form ... [ ]
D--Delete Save field ..... [ ]
   Form ..... [ ]
C--Copy new form name ..... [ ]
   from form ..... [ ]
   from Forms File (opt) ..... [ ]
X--Compile Forms File
   Optional: Fast Forms File [ ]
R--Relate child form [ ] to parent form [ ]
PREV  NEXT  REFRESH  PREV  NEXT  MAIN/  EXIT
FORM  FORM  FORM      FORM  FORM  RESUME  FORM
```

Figure 3-2. The Main Menu

## Select the GO TO FORM Option

FORMSPEC opens the ORDFORM1 forms file and issues the Main Menu, as shown in Figure 3-2. Notice that the name of the open forms file (ORDFORM1) is shown in the window line after the FORMS FILE: heading.

### Action

Type G in the selection box to indicate that you want to "Go To" one of the three choices listed. Since you need to modify the SHIPTO form, you type the name of the form (SHIPTO) in the Go to form field. Press ENTER.

If you want to modify a specific field, you also specify the field name in the field labeled field. In this exercise, you modify the form first before changing field specifications, so you do not enter a field name. If you only type G in the selection box and do not specify a form or field, the Globals Menu is displayed, which is described in Section 7.

```

FORMSPEC v.uu.ff Form Menu                                FORMS FILE: ORDFORM1

Form Name [ORDER ]
Repeat Option [N]
             N--No Repeat
             A--Repeat, appending
             R--Repeat, overlaying

Next Form [C]                                         Name [$HEAD ]
             C--Clear before Next Form
             A--Append Next Form
             F--Freeze, then append Next Form

Function Key Label [ ]
             Y--Define form level function key labels.

Reproduced from [ ] (opt)

Comments [Only 1 form in this file. ]

PREV FORM  NEXT FORM  REFRESH  PREV  NEXT  MAIN/RESUME  EXIT

```

Figure 3-3. The Form Menu

## Change Form Name

Since you selected to go to the SHIPTO form at the Main Menu, when you press ENTER FORMSPEC issues the Form Menu for the SHIPTO form, which is a duplicate of the form shown earlier as Figure 2-4. Since the SHIPTO form is to be changed significantly in this section by the addition of order information, change the form name to ORDER.

### Action

Type ORDER in the Form Name field, as shown in Figure 3-3. No other changes are needed. Press ENTER.

The form name, previously SHIPTO, is now ORDER.

```

-
*****ABC MANUFACTURING*****
                                     Date: [ordate           ]

SHIP TO:
  Name [name                        ]
Address [address                    ]
  City [city                        ] State [st]
  Zip [zip          ] Telephone [phone      ]

PREV  NEXT  REFRESH  PREV  NEXT  MAIN/  EXIT
FORM  FORM  FORM     FORM  FORM  RESUME FORM

```

Figure 3-4. The Form Layout for the ORDER Form

## Add New Fields

FORMSPEC displays the Form Layout Menu for the ORDER form (which was called SHIPTO), as shown in Figure 3-4. Once a form is displayed, you can change the placement, name or size of any existing fields, or the header information and text on the form.

### Action

For this exercise, you add six new fields to the ORDER form; the lengths and positions of the new fields are defined in Table 3-1.

Table 3-1. Defining the Fields for the ORDER Form

Field	Tag	Line No.	Length
Order Number	ordate	5	6
Quantity	qty	17	5
Part Number	partnum	17	9
Description	description	17	43
Unit Price	uprice	17	7
Total Price	tprice	17	7

```

-
                                     *****ABC MANUFACTURING*****

Order Number: [ordnum]                               Date: [ordate           ]

SHIP TO:
  Name [name                ]
  Address [address          ]
  City [city                ] State [st]
  Zip [zip                  ] Telephone [phone            ]

QTY   PART NO.   DESCRIPTION                               UNIT   TOTAL
qtty.. partnum.. description.....                uprice. tprice.

PREV  NEXT  REFRESH  PREV  NEXT  MAIN/  EXIT
FORM  FORM  FORM      FORM  FORM  RESUME FORM

```

Figure 3-5. The Form Layout for the ORDER Form

## Use Non-Printing Delimiters

The result is an order number field at the top of the form and five fields of order information at the bottom, as shown in Figure 3-5.

### Action

In order to fit these five new fields on a single line, you need to use nonprinting delimiters, which are created by pressing ESCAPE and then [ to indicate the start-of-field; pressing ESCAPE followed by ] to indicate the end-of-field.

The start-of-field delimiter is in the same position as the first character in the field, and the end-of-field delimiter is in the same position as the last character. Thus, these delimiters, unlike the brackets, take up no room on the screen. You use the field edit keys, including Insert char and Delete char, to correct the position of the nonprinting delimiters.

FORMSPEC allows you to fill a field with dots (periods) in any spaces left in a field after the field tag. This is particularly handy when you use nonprinting delimiters instead of brackets because the dots clearly indicate the length of each field. Like the field tag, these dots are not shown when the form is displayed for the user; they are a convenience only to the forms designer.

## Action

Using the terminal field edit and cursor control keys changes the existing form to look like Figure 3-5. When the form is complete, press ENTER.

The new fields are recorded as part of the ORDER form in the ORDFORM1 forms file. Remember that this form cannot be used by ENTRY until the forms file is recompiled.

```
FORMSPEC v.uu.ff Field Menu                                FORM: ORDER
*****ABC MANUFACTURING*****

Order Number: [ordnum]                                     Date: [ordate   ]
                ^
Num [8  ] Len [6  ] Name [ORDNUM   ] Enh [HI  ] FType [0] DType [CHAR]
Initial Value [                                     ]

*** Processing Specifications ***
```

```
PREV  NEXT  FIELD  REFRESH          PREV  NEXT  MAIN/  EXIT
FORM  FORM  TOGGLE  REFRESH          FORM  FORM  RESUME  FORM
```

Figure 3-6. The Field Menu for the Order Number Field

## Examine the Field Menu

When you press ENTER to record the completed form, FORMSPEC automatically issues the Field Menu for ORDNUM, the first field. The menu shows the field with all default values: enhancement is HI for Half-bright, Inverse video; field type is 0 for Optional; and data type is CHAR for any Character.

Note that the field number for the ORDNUM field is 8 even though it is the first field in physical screen order and precedes the ORDATE field, which is field number 1. This is because field numbers are assigned in the order in which fields are created. These numbers are not related to the order of the fields on the form.

Field names are shown on the Field Menu in all uppercase letters, whereas

the field tag entered during form layout can be any combination of upper or lower case letters, such as `ordnum`. The uppercase name, `ORDNUM`, is the name by which the field is identified on the Field Menu, as shown in Figure 3-6. If you want, you can change the field name to another all uppercase field name. You may want to do so for several reasons. For instance, since lowercase letters are never used in the field name, but only appear in the field tags, the shift to uppercase may force you to assign a new name to a field so that the field names are unique. Thus, if you layout a form with two fields tagged respectively, `F1` and `f1`, the field names displayed on the Field Menus are no longer unique; they both appear as `F1`. You must make them unique, using new field names of up to 15 characters long for one or both of the fields. You might also change a field name to make it more meaningful. For example, suppose the name `TR` identifies a two-character field. Changing it to `TAX_RATE` makes its function easier to remember.

## **Topic: Field Menu Fields**

The next step in designing this form is to specify the field edits to be used for each field of this form Figure 3-6.

But before doing this, read the following brief description of the field attributes issued on all Field Menus.

### **The Field Number Field**

A field number is assigned by `FORMSPEC` to every field. This number is initially based on the position of the field on the form (for example, `ordate` is field number 1), and this number never changes. If you change the form layout at a later time so that `ordate` is no longer the first field, it is still identified as field number 1. (Note that these field numbers are not significant unless you use the `VPLUS/V` intrinsics to write your own application or to modify `ENTRY`.)

### **The Field Length Field**

Field length, like the field number, is based on the form layout. The number of characters you leave between the field delimiters is shown on the Field Menu as the length of that field. Unlike the field number, the length changes whenever you change the field size on the form layout. In our example, 18 character positions were allowed for the date, so `ordate` has a field length of 18.

### **The Field Name Field**

Field tags are used when laying out the form--field names are used on the Field Menu to uniquely identify the field. When the Field Menu is initially displayed, the field tag is used as the field name by default and is always shown in uppercase letters. This is because `FORMSPEC` automatically shifts any lowercase letters in the field tag (the name between field delimiters on the form) to uppercase. It does not change

any letters outside the field delimiters. Both upper and lowercase letters are allowed during form layout so that you can have up to 52 one-character fields. In the current example, the field tag `ordate` was shifted up to become the official field name `ORDATE`.

### **The Field Enhancement Field**

The field enhancement indicates how the field looks when `ENTRY` displays the form. Unless you specifically change the field enhancement in the Field Menu, the default enhancement is `HI` meaning that the field will be shown as Half-bright and Inverse video, which distinguishes them from all

other text on the form. (When you run `ENTRY`, you will see what this means.) This is the default enhancement for all fields unless you change it. Other enhancements could be `U` for Underline, `B` for Blinking, `S` for Security, `NONE` for no enhancement, and `1 - 8` for Color. Combinations of up to four characters of the possible enhancements are also allowed, such

as `HIU` for a underlined, half-bright, inverse video field or `B2` for a red blinking field.

### **The Field Type Field**

The field type determines whether the user can leave data out of the field (`O` for optional), must enter data (`R` for required), or cannot enter data (`D` for display-only), in which case the application must supply the data. There is also a field type which forces the system to process the field even if it is blank (`P` for process). Unless you change the field type, all fields are optional.

### **The Data Type Field**

The data type determines the validity of data that is entered in the field, the formatting method allowed, and the operations, if any, allowed

on the field. Each field must be specified as one of three types: character, date, or numeric data. By default, all fields are type `CHAR`, which means you can enter any characters in the field. The other two types indicate how dates or numbers can be entered and are described later in this section.

### **The Initial Value Field**

If you want, you can assign a value to appear in the field when the form is first displayed by `ENTRY`.

```

FORMSPEC v.uu.ff Field Menu                                FORM: ORDER
*****ABC MANUFACTURING*****

Order Number: [ordnum]                                     Date: [ordate   ]
                ^
Num [8  ] Len [6  ] Name [ORDNUM   ] Enh [HI  ] FType [R]  DType [CHAR]
Initial Value [100000]                                     ]
-----
*** Processing Specifications ***

```



Figure 3-7. The Field Menu for the Order Number Field

## Specify Field Type and Initial Value

### Action

Since you want to make sure the user always enters an order number, change the field type from O for optional to R for required on the Field Menu for ORDNUM.

This makes it impossible for the user to leave the order number blank when ENTRY displays the ORDER form.

### Action

Let's say that the order number must always be a number with a value of at least 100000. Enter this value as an initial value on the Field Menu for ORDNUM.

This initial value appears in the field when the form is first displayed by ENTRY. The user can then change the initial value or leave it as is. (Later, Section 4 shows how to have this initial value incremented automatically to insure that each order has an ascending and unique order number.)

### Action

When these changes are made to the Field Menu for ORDNUM, the menu appears as shown in Figure 3-7. Press ENTER.

**Table 3-2. Simple Field Edits for the ORDER Form**

Field	Name	Simple Edit
Order Number	ORDNUM	Change Field Type (FType) to R for required. Add Initial Value of 100000.
Date	ORDATE	Change Field Type (FType) to R for required. Change Data Type (DType) to MDY for a date in the order: Month Day Year.
Quantity	QTY	Change Field Type (FType) to R for required. Change Data Type (DType) to DIG for all digits.
Part Number	PARTNUM	Change Field Type (FType) to R for required.
Unit Price	UPRICE	Change Field Type (FType) to R for required. Change Data Type (DType) to NUM2 for a numeric value with two decimal positions.
Total Price	TPRICE	Change Field Type (FType) to R for required. Change Data Type (DType) to NUM2 for a numeric value with two decimal positions.

## Specify Field Edits

When you press ENTER to record the field type specification on the Field Menu for ORDNUM, FORMSPEC issues the next sequential Field Menu, which is the one for ORDATE. The menu for the ORDATE field also has only default specifications.

```
FORMSPEC v.uu.ff Field Menu                                FORM: ORDER
*****ABC MANUFACTURING*****

Order Number: [ordnum]                                     Date: [ordate      ]
                                                         ^
Num [1  ] Len [18  ] Name [ORDATE      ] Enh [HI  ] FType [R] DType [MDY ]
Initial Value [
-----
*** Processing Specifications ***

PREV  NEXT  FIELD  REFRESH          PREV  NEXT  MAIN/  EXIT
FORM  FORM  TOGGLE                FORM  RESUME
```

Figure 3-8 The Field Menu for the Order Number

### Action

Change the field type to R for required and change the data type to MDY.

When a date is specified as MDY, this means the user must enter a valid date in the order: Month Day Year.

### Action

Press ENTER to record the changed field attributes.

FORMSPEC issues the Field Menu for NAME, as shown in Figure 3-9.

```

FORMSPEC v.uu.ff Field Menu                                FORM: ORDER
SHIP TO:
  Name [name                ]
  ^
Num [2 ] Len [42 ] Name [NAME ] Enh [HI ] FType [0] DType [CHAR]
Initial Value [                ]
*** Processing Specifications ***

PREV FORM  NEXT FORM  FIELD TOGGLE  REFRESH          PREV  NEXT  MAIN/RESUME  EXIT

```

Figure 3-9 The Field Menu for NAME

**Action**

Let's keep the default values for the name, address, and phone number fields, meaning they are all to be optional and, if specified, to have any characters. To do this, just skip these fields by pressing NEXT at each menu until the Field Menu for QTY is issued.

```

FORMSPEC v.uu.ff Field Menu                                FORM: ORDER
QTY  PART NO.      DESCRIPTION                                UNIT  TOTAL
                                           PRICE  PRICE
qty.. partnum..  description.....  uprice. tprice.
^
Num [9 ] Len [5 ] Name [QTY ] Enh [HI ] FType [R] DType [DIG ]
Initial Value [                ]
*** Processing Specifications ***

PREV FORM  NEXT FORM  FIELD TOGGLE  REFRESH          PREV  NEXT  MAIN/RESUME  EXIT

```

Figure 3-10. The Field Menu for the Quantity Field

The QTY field is the first field in the line of order information, as shown in Figure 3-10. This field should always be specified by the user and the quantity should be entered as a number. To insure that an unsigned number, consisting only of the digits 0-9, is entered in this field, use the data type DIG.

**Action**

Change the field type to R for required and change the data type to DIG. Press ENTER to record the changes.

```

FORMSPEC v.uu.ff Field Menu                                FORM: ORDER
QTY  PART NO.      DESCRIPTION                                UNIT  TOTAL
                                           PRICE  PRICE
qty.. partnum..  description.....  uprice. tprice.
  ^
Num [10 ] Len [9  ] Name [PARTNUM ] Enh [HI ] FType [R] DType [CHAR]
Initial Value [

```

---

\*\*\* Processing Specifications \*\*\*

PREV	NEXT	FIELD	REFRESH		PREV	NEXT	MAIN/ RESUME	EXIT
FORM	FORM	TOGGLE						

Figure 3-11. The Field Menu for the Part Number Field

```

FORMSPEC v.uu.ff Field Menu                                FORM: ORDER
QTY  PART NO.      DESCRIPTION                               UNIT  TOTAL
                                           PRICE  PRICE
qty.. partnum..  description.....  uprice.  tprice.
      ^
Num [11 ] Len [43 ] Name [DESCRIPTION ] Enh [HI ] FType [0] DType [CHAR]
Initial Value [

```

---

\*\*\* Processing Specifications \*\*\*

PREV FORM	NEXT FORM	FIELD TOGGLE	REFRESH		PREV	NEXT	MAIN/ RESUME	EXIT
--------------	--------------	-----------------	---------	--	------	------	-----------------	------

**Figure 3-12. The Field Menu for the Description Field**

**Action**

Change the Field Menu for PARTNUM to a required field, as shown in Figure 3-11. Press ENTER to record the change.

This means that the user must enter a part number.

**Action**

When the Field Menu for DESCRIPTION is displayed, as shown in Figure 3-12, press NEXT.

This keeps the defaults, which specify that the field is optional and able to accept any characters. Pressing NEXT rather than ENTER on Field Menus that do not require any changes is a good habit to build. This keeps the forms file size to a minimum.

```

FORMSPEC v.uu.ff Field Menu                                FORM: ORDER
QTY  PART NO.      DESCRIPTION                                UNIT  TOTAL
                                           PRICE  PRICE
qty.. partnum..  description.....  uprice. tprice.
                                           ^
Num [12 ] Len [7 ] Name [UPRICE ] Enh [HI ] FType [R] DType [NUM2]
Initial Value [ ]

```

---

\*\*\* Processing Specifications \*\*\*

PREV FORM	NEXT FORM	FIELD TOGGLE	REFRESH		PREV	NEXT	MAIN/ RESUME	EXIT
--------------	--------------	-----------------	---------	--	------	------	-----------------	------

Figure 3-13. The Field Menu for the Unit Price Field

```

FORMSPEC v.uu.ff Field Menu                                FORM: ORDER
QTY  PART NO.      DESCRIPTION                                UNIT  TOTAL
                                           PRICE  PRICE
qty.. partnum..  description.....  uprice. tprice.
                                           ^
Num [12 ] Len [7 ] Name [UPRICE ] Enh [HI ] FType [R] DType [NUM2]
Initial Value [ ]

```

---

\*\*\* Processing Specifications \*\*\*

PREV FORM	NEXT FORM	FIELD TOGGLE	REFRESH		PREV	NEXT	MAIN/ RESUME	EXIT
--------------	--------------	-----------------	---------	--	------	------	-----------------	------

Figure 3-14. The Field Menu for the Total Price Field

The next two fields are UPRICE, as shown in Figure 3-13, and TPRICE, as shown in Figure 3-14. The user needs to enter prices into these fields.

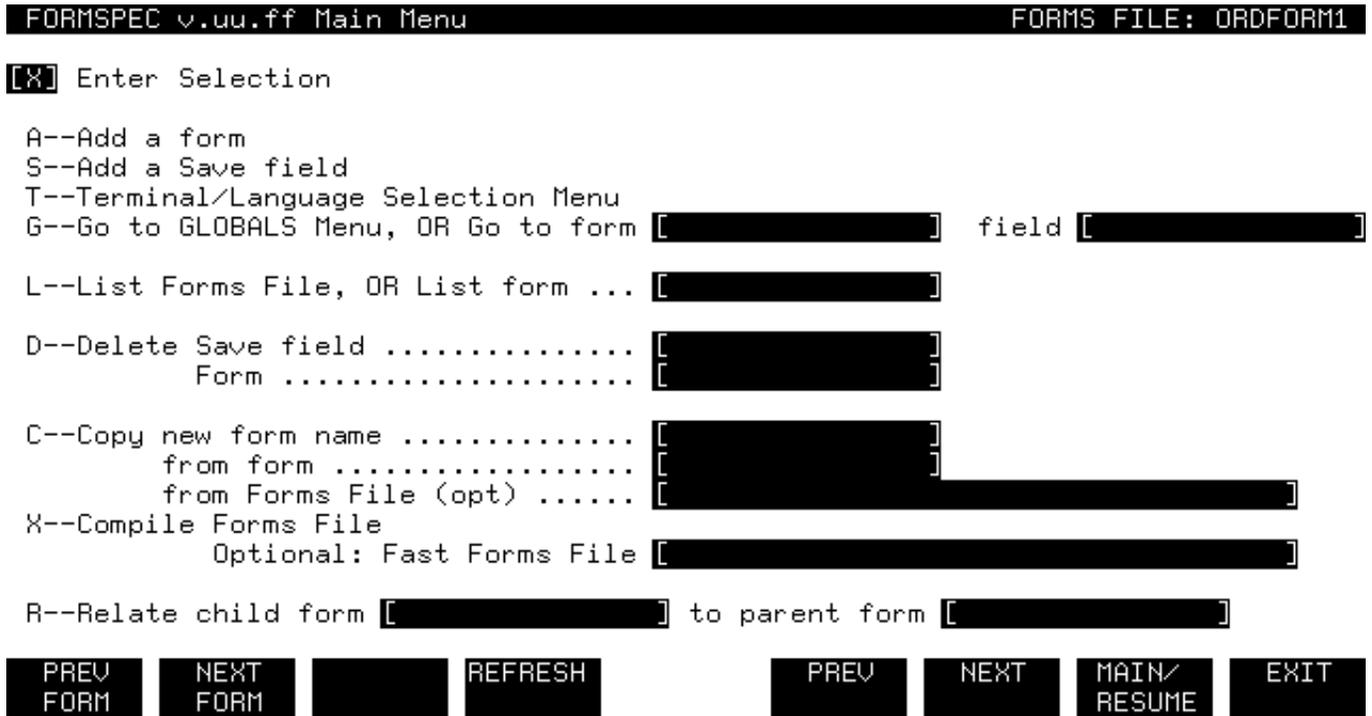
**Action**

Change the field type of each field to R for required and change the data type of each field to NUM2. Press ENTER for each Field Menu to record these changes.

These are all the changes to be made in this section.

**Action**

Press MAIN/RESUME to return to the Main Menu.



07:13  
**Figure 3-15. The Main Menu**

**Action**

Type X in the selection box, as shown in Figure 3-15. Press ENTER to compile ORDFORM1. When the compile is complete, press EXIT to exit FORMSPECC.

At this point, the forms file ORDFORM1 with the new ORDER form is compiled and can be used for order entry through ENTRY. The fields are edited as shown in Table 3-8. The next step is use ENTRY to enter data into a batch file using the new form.



```
:RUN ENTRY.PUB.SYS
```

```
HP32209v.uu.ff ENTRY (C) HEWLETT-PACKARD CO. 1986
```

```
ENTER FORMS FILE NAME AND PRESS RETURN: ORDFORM1
```

```
ENTER BATCH FILE NAME AND PRESS RETURN: ORDATA
```

**Figure 3-16. Run ENTRY and the ORDER Form, Batch Record No. 1**

## Enter and Review Data with ENTRY

In this exercise, using ENTRY, you enter data into the new ORDER form in the ORDFORM1 forms file.

### Action

Run ENTRY, as shown in Figure 3-16. When the prompts are issued, type the name of the newly compiled forms file, ORDFORM1, and use ORDATA for the batch file.

Because the ORDFORM1 file has been changed significantly and recompiled, you should not use the batch file (CUSTLIST) that was used to collect data in Section 2. Instead, you use a different batch file for collecting order information by specifying a different name (ORDATA) as shown. If you had entered the name of an existing batch file previously used for data collection with a forms file that has been recompiled, a message would be issued:

```
WARNING: Forms file recompiled since this batch was created.  
Enter "Y" to continue:
```

This message is issued to insure that you do not unwittingly add data to a file that already contains data in a different format. Since the ORDATA batch file is new, this message is not issued.

When the files have been named, ENTRY displays the form, as shown in Figure 3-16. Note that all fields are blank, except for the Order Number field that contains the initial value 100000.

```

-
                *****ABC MANUFACTURING*****
Order Number: [100000]                Date: [          ]

SHIP TO:
  Name [          ]
  Address [          ]
  City [          ] State [  ]
  Zip [          ] Telephone [          ]

QTY   PART NO.   DESCRIPTION                               UNIT   TOTAL
      [          ] [          ] [          ]          PRICE  PRICE

ENTRY v.uu.ff          Batch Record #2          Mode: Collect
HEAD [          ] PRINT [          ] REFRESH [          ] NEXT [          ] BROWSE [          ] EXIT [          ]
FORM [          ] [          ] [          ] [          ] FORM [          ] [          ] [          ]

```

Figure 3-17. The ORDER Form, Batch Record No. 1

## Change Initial Value

You can change the initial value by typing a new value over the value that appears in the field. However, it is more likely that you need to change only the last two or three digits of this value. To do this, use a cursor control key to place the cursor under the first digit you want to change and then type in the new characters.

### Action

Change the order number to 100001 by changing the last digit to 1, as shown in Figure 3-17.

```

-
*****ABC MANUFACTURING*****
Order Number: [100001] Date: [30 SEP 1986 ]

SHIP TO:
Name [FOLEY BOWMAN, INC. ]
Address [369 Marietta Way ]
City [Los Angeles ] State [Ca]
Zip [90020 ] Telephone [(213) 478-9654]

QTY PART NO. DESCRIPTION UNIT TOTAL
PRICE PRICE
[ ] [PM391-001] [ ] [10.98] [21.96]

The field must be a valid date, in MDY order.
HEAD | PRINT | REFRESH | NEXT | BROWSE | EXIT
FORM | | | FORM | |

```

Figure 3-18. The ORDER Form, Batch Record No. 1 with Errors

## Error Handling

### Action

Type data in the form, as shown in Figure 3-18. Press ENTER.

ENTRY does not accept the data and issues the message shown in the window line, which is used for error as well as status messages. The reason for the error message is that the date is in DMY order, rather than in MDY order. Since the field is defined as an MDY type, the month must precede the day of the month. Note that if the month is misspelled, the same error message is issued. No message is ever issued when the data type is CHAR since that type accepts any characters.

To help you correct the field in error, the cursor is positioned to the beginning of the date field so that you can correct this field with no extra effort. The field itself is underlined and full bright, rather than half bright, to distinguish it from the correct fields.

### Action

Change the date to SEP 30 1986. Press ENTER.

```

-
*****ABC MANUFACTURING*****

Order Number: [100001]           Date: [SEP 30 1986 ]

SHIP TO:
  Name [FOLEY BOWMAN, INC. ]
  Address [369 Marietta Way ]
  City [Los Angeles ] State [Ca]
  Zip [90020 ] Telephone [(213) 478-9654]

QTY   PART NO.   DESCRIPTION                               UNIT   TOTAL
      PM391-001  [REDACTED]                               PRICE PRICE
[REDACTED] [REDACTED] [REDACTED]          10.98  21.96

The required field is empty.
HEAD  PRINT  REFRESH  NEXT  BROWSE  EXIT
FORM  [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED] [REDACTED]

```

Figure 3-19. The ORDER Form, Batch Record No. 1

## Multiple Errors

The data is still not accepted, as shown in Figure 3-19, because a value for the Quantity field is omitted. This is not allowed because the field was defined as a required field. Note that the field is underlined and full bright to let you know that it is in error.

### Action

Enter the value 2 in the Quantity field. Press ENTER.

Since all fields with errors are enhanced (in this case they are underlined and full bright as you saw in the form that corresponds to Figure 3-19), you could have corrected the Quantity field when you corrected the date. If you had, no message would have been issued for the Quantity error. One message at a time is displayed when there is more than one error. The displayed message describes the error for the first incorrect field in screen order. When that field is corrected and you press ENTER, a message describing the next field with an error is displayed. If all fields with errors are corrected at once, no further message is displayed after the first message.

### Action

If Record #1 is still displayed, check if there are any other fields in

error. If so, correct the errors and press ENTER.

```
-
                               *****ABC MANUFACTURING*****
Order Number: [100000]           Date: [                ]

SHIP TO:
Name [                ]
Address [                ]
City [                ] State [  ]
Zip [                ] Telephone [                ]

QTY  PART NO.  DESCRIPTION  UNIT PRICE  TOTAL PRICE
[    ] [    ] [                ] [    ] [    ]

ENTRY v.uu.ff  Batch Record #2  Mode: Collect
HEAD          PRINT  REFRESH          NEXT  BROWSE  EXIT
FORM          [    ] [    ] [    ] [    ] [    ] [    ]
```

Figure 3-20. The ORDER Form, Batch Record No. 2

## Enter Data

This time the data is accepted, and Record #1 is written to the batch file ORDATA. The ORDER form is displayed, ready for Record #2, as shown in Figure 3-20. Note that all fields are blank, except for the Order Number field that contains the initial value 100000.

### Action

Continue to enter data in this form until there are four records in the batch file ORDATA.

```

-
*****ABC MANUFACTURING*****
Order Number: [100001] Date: [SEP 30 1986 ]

SHIP TO:
Name [FOLEY BOWMAN, INC. ]
Address [369 Marietta Way ]
City [Los Angeles ] State [CA]
Zip [90020 ] Telephone [(213) 478-9654]

QTY PART NO. DESCRIPTION UNIT TOTAL
PRICE PRICE
2 PM391-001 10.98 21.96

ENTRY v.uu.ff Batch Record #1 Mode: Browse
FIRST DELETE PRINT REFRESH PREV NEXT COLLECT EXIT
REC REC REC REC REC REC REC REC

```

Figure 3-21. The ORDER Form, Batch Record No. 1

## Change to Browse Mode

If you want to look at the data entered so far in the batch file, you use the Browse mode available with ENTRY.

### Action

Press BROWSE.

This key places ENTRY in "Browse/Modify" mode. In order to examine the data in the entire file, you start with the first record and step through the records in the batch file in the sequence in which they were entered.

### Action

Press FIRST REC.

The first record in the file is displayed exactly as it was entered, as shown in Figure 3-21. That is, ENTRY paints the form on the screen and displays the data that you entered in the form as Record #1.

```

-
*****ABC MANUFACTURING*****
Order Number: [100002] Date: [SEP 30 1986 ]

SHIP TO:
Name [ABRAHAM GRANDVIEW ]
Address [19831 Sloan Street ]
City [Los Angeles ] State [CA]
Zip [90120 ] Telephone [(213) 323-1654]

QTY PART NO. DESCRIPTION UNIT TOTAL
PRICE PRICE
4 PC021-10A 15.00 60.00

ENTRY v.uu.ff Batch Record #2 Mode: Browse
FIRST DELETE PRINT REFRESH PREV NEXT COLLECT EXIT
REC REC REC REC REC REC REC REC

```

Figure 3-22. The ORDER Form, Batch Record No. 2

## Browse the Records

### Action

To look at the next record, press NEXT REC.

Record #2 is displayed, such as the example in Figure 3-22. Your Record #2 contains the data you invented and entered earlier in this exercise.

### Action

Continue pressing NEXT REC until all the records in the file have been displayed.

When there are no more records, ENTRY displays the following message in the window line:

There are no more batch records.

If you want to display any records while ENTRY is still in Browse mode, you press PREV REC. This function operates only in Browse mode; if you have returned to Collect mode, it has no meaning.

\*\*\*\*\*ABC MANUFACTURING\*\*\*\*\*

Order Number: [100000]

Date: [ ]

SHIP TO:

Name [ ]  
Address [ ]  
City [ ] State [ ]  
Zip [ ] Telephone [ ]

QTY	PART NO.	DESCRIPTION	UNIT PRICE	TOTAL PRICE
[ ]	[ ]	[ ]	[ ]	[ ]

ENTRY v.uu.ff Batch Record #5 Mode: Collect  
HEAD PRINT REFRESH NEXT BROWSE EXIT  
FORM FORM

Figure 3-23. The ORDER Form, Batch Record No. 5

## Return to Collect Mode

Now, you are ready to return to Collect mode and continue data collection.

### Action

Press COLLECT.

Assuming Record #4 is the last record in the file, the form with only initial values is displayed so you can continue entering data into Record #5, as shown in Figure 3-23.

Remember, the mode in which ENTRY is currently operating is always displayed in the window line.

### Action

When you are through entering data, either in Collect or Browse mode,

press EXIT.

When you press EXIT in either mode, ENTRY terminates and returns you to MPE. The next time you run ENTRY with the *same* batch file (assuming you have not recompiled the forms file) you enter data into the record following the last record in the file.

## Chapter 4 Full Field Edits

This section introduces the simple programming language provided by FORMSPEC, called **processing specifications**. This language includes edit statements that are used for full field editing, the subject of this section. It also includes other types of statements that are used for more advanced processing, as you will learn in later sections.

The edit statements consist of a statement name followed by one or more parameters. They are entered in the Processing Specifications area of the Field Menu and determine precisely how ENTRY tests any data entered into that field. As shown in Section 3, if the data entered in a field fails a simple field edit, ENTRY returns a generic error message. However, the precision provided by edit statements requires more specific error messages. FORMSPEC allows you to specify a custom error message for each edit statement which lets the user know exactly what the problem is and how to correct it.

The ORDFORM1 forms file contains one form (ORDER) that now has simple data type and field type edits. In this section, you will learn:

### Using FORMSPEC:

- \* What the available edit statements are
- \* How to specify custom error messages
- \* Locate field to change (Main Menu)
- \* Use FIELD TOGGLE (Field Menu)
- \* Enter processing specifications (Field Menu):

- \* Single value comparisons (EQ,GE)
- \* Range checks (IN,NIN)
- \* Test for minimum length (MINLEN)
- \* Table checks (IN,NIN)
- \* Pattern match (MATCH)
- \* Custom error messages

### Using ENTRY

- \* Display custom error messages
- \* Use REFRESH to clear data from form
- \* Modify data in Browse mode

---

:RUN FORMSPEC.PUB.SYS

---

```

-
*****ABC MANUFACTURING*****
Order Number: [100000] Date: [ ]

SHIP TO:
Name [ ]
Address [ ]
City [ ] State [ ]
Zip [ ] Telephone [ ]

QTY PART NO. DESCRIPTION UNIT TOTAL
PRICE PRICE

ENTRY v.uu.ff Batch Record #5 Mode: Collect
HEAD | PRINT | REFRESH | NEXT | BROWSE | EXIT
FORM | | | FORM | |

```

Figure 4-1. The Forms File Menu

## Run FORMSPEC

The overall form layout and specification of the ORDER form is not changed. However, field edits need to be added to several fields in the form.

### Action

Issue the RUN command, as shown in Figure 4-1. Press RETURN.

Specify the forms file (ORDFORM1) on the Forms File Menu, also shown in Figure 4-1. Press ENTER.

```

FORMSPEC v.uu.ff Main Menu                                FORMS FILE: ORDFORM1
[G] Enter Selection
Forms file is 1% full
A--Add a form
S--Add a Save field
T--Terminal/Language Selection Menu
G--Go to GLOBALS Menu, OR Go to form [ORDER ] field [ORDATE ]
L--List Forms File, OR List form ... [ ]
D--Delete Save field ..... [ ]
   Form ..... [ ]
C--Copy new form name ..... [ ]
   from form ..... [ ]
   from Forms File (opt) ..... [ ]
X--Compile Forms File
   Optional: Fast Forms File [ ]
R--Relate child form [ ] to parent form [ ]
PREV  NEXT  REFRESH  PREV  NEXT  MAIN/  EXIT
FORM  FORM  FORM      FORM  FORM  RESUME

```

Figure 4-2. The Main Menu, Selecting Go To Field

## Select GO TO FIELD Option

When you press ENTER at the Forms File Menu, the Main Menu is issued. Since there are no changes to the Form Menu or to the form layout, you can go directly to the first Field Menu to which an edit is to be added. Suppose this Field Menu is for the date field.

### Action

Type G in the selection box, the form name ORDER in the form name box, and the field name ORDATE in the field box (shown in Figure 4-2). Press ENTER.

```

FORMSPEC v.uu.ff Field Menu                                FORM: ORDER
*****ABC MANUFACTURING*****

Order Number: [ordnum]                                Date: [ordate ]
                                         ^
Num [1 ] Len [18 ] Name [ORDATE ] Enh [HI ] FType [R] DType [MDY ]
Initial Value [ ]
-----
*** Processing Specifications ***
-

```

```

PREV  NEXT  FIELD  REFRESH
FORM  FORM  TOGGLE

PREV  NEXT  MAIN/  EXIT
FORM  FORM  RESUME

```

Figure 4-3. The Field Menu for ORDATE in the ORDER Form

## The Field Toggle

When you press ENTER, FORMSPEC displays the Field Menu for ORDATE, as shown in Figure 4-3. All the field edits described in this section, as well as the advanced processing specifications described in later sections, are entered in the area of the Field Menu below the heading:

```
*** Processing Specifications ***
```

(The processing specifications are statements whose syntax is fully described in the *VPLUS/V Reference Manual*.) In order to enter these statements, you must press the function key that corresponds to the FIELD

TOGGLE label. This key signals to FORMSPEC that you want to enter statements in the lower area of the Field Menu, and causes the cursor to be positioned to the first character position (top left) of this area, as shown in Figure 4-3.

### Action

Press FIELD TOGGLE.

When you press FIELD TOGGLE to enter processing specifications, FORMSPEC changes the mode of operations from format mode to unformatted mode. In unformatted mode, you can type anywhere on the screen, whereas in formatted mode (the normal mode for FORMSPEC) you are restricted to

particular fields that are unprotected. You can now enter any edit (or processing) statements that apply to the current field. If you are in the lower, processing specification, part of the menu and want to change an attribute of the upper part, you should always press FIELD TOGGLE to return to the upper part and then make your change.

## Topic: Edit Statements

The next step in designing this form is to specify full field edits using edit statements, which are part of the FORMSPEC Processing Specifications language. Edit statements enable you to specify more precise field edits than the simple edits you specified in Section 2. Before you specify full edits in the exercise, read the following brief description of the edit statements available with FORMSPEC.

### Types of Statements

Edit statements consist of statement names followed by parameters, which are values or patterns that you specify. You also have the option of specifying an error message for each edit statement to be used if the field fails the test. Following are descriptions of the edits available.

**Minimum Length.** The value entered must be at least a specified number of characters long. For example: MINLEN 10 means at least ten characters must be entered in the field.

**Single Value Comparison.** Single value comparisons are a set of edit statements, shown in Table 4-1, that compare the value entered in the current field (the field whose Field Menu has the edit statement) with a value specified in the statement. The value to which the entered data is compared must always be the same data type as the field (character, numeric, or date). Depending on which of the statement names you use, the value entered must be equal to, equal to or greater than, equal to or less than, less than, greater than, or not equal to a specified value.

For example: GE FIELD2 means the value entered in the current field must be equal to or greater than the value entered in a different field, FIELD2.

**Table 4-1. Comparison Editing Statements**

Statement	Meaning
EQ	is equal to
NE	is not equal to
LT	is less than
GT	is greater than
LE	is less than or equal to
GE	is greater than or equal to

**Range Check.** Another type of edit statement specifies a range of values within which the entered value must fall. A range statement consists of IN (*is in*) or NIN (*is not in*) followed by a range of values. The first value in the range is separated from the last by a colon (:). The first value must be the lower value.

Possible range checks are:

- IN "AAMES":"AZIMOV"                      Only names between AAMES and AZIMOV (inclusive) are acceptable in a character type field.
- IN \$TODAY: !12/31/86!                      Only dates between today's date through 1986 can be entered.
- NIN 20:50                                      Values between 20 and 50 are treated as errors. (Note that this statement indicates a range of unacceptable values.)
- IN 10:20, F1:F4, 100:300                      Means the value must be between 10 and 20, or between the values entered in fields F1 and F4, or between 100 and 300, inclusive.

**Table Check.** A special case of the range check is a table (or list) check. This edit determines whether the entered value is one of a set of values arranged as a table or list. Like the range check, the table check statement starts with either IN (*is in*) or NIN (*is not in*). Unlike a range check, every value in the table must be listed and separated by commas. For example, consider the following table checks:

- IN 2,4,6,8,10,12,14,16,20                      Entered value must be one of the specified values.
- IN 5,10,15,F7+5                                      Means the value must be 5, 10, 15 or the current value of field F7 plus 5.



```

FORMSPEC v.uu.ff Field Menu                                FORM: ORDER
*****ABC MANUFACTURING*****

Order Number: [ordnum]                                     Date: [ordate      ]
                                                ^
Num [1 ] Len [18 ] Name [ORDATE ] Enh [HI ] FType [R] DType [MDY ]
Initial Value [ ]
-----
*** Processing Specifications ***
EQ $TODAY

```

```

PREV  NEXT  FIELD  REFRESH
FORM  FORM  TOGGLE

PREV  NEXT  MAIN/  EXIT
FORM  FORM  RESUME

```

Figure 4-4. The Field Menu for ORDATE in the ORDER Form

## Use Single Value Comparisons

Now that you understand how to use the FIELD TOGGLE, lets add some processing specifications to the field. The first type of field edit you will use is a single value comparison.

### Action

Press FIELD TOGGLE. Type the field edit statement EQ \$TODAY in the lower menu, as shown in Figure 4-4. Press ENTER.

The statement EQ \$TODAY is one possible comparison. It means that the data entered in the field must be today's date. Other single value comparisons you could enter in a date field include:

- |             |  |
|-------------|--|
| GE !1/1/86! | Entered date must be in 1986 or a subsequent year. |
| LT \$TODAY  | Entered date must be earlier than today's date.    |

```

FORMSPEC v.uu.ff Field Menu                                FORM: ORDER
QTY  PART NO.      DESCRIPTION                               UNIT  TOTAL
                                           PRICE  PRICE
qty.. partnum..  description.....  uprice. tprice.
^
Num [9 ] Len [5 ] Name [QTY ] Enh [HI ] FType [R] DType [DIG ]
Initial Value [
*** Processing Specifications ***
GE 1

```

```

PREV  NEXT  FIELD  REFRESH          PREV  NEXT  MAIN/  EXIT
FORM  FORM  TOGGLE  REFRESH          FORM  RESUME  EXIT

```

**Figure 4-5. Field Menu for QTY in the ORDER Form**

Single value comparisons can also be used for other fields of the form. If you want to ensure that the number of parts ordered is one or more, you use the GE statement to test that the data entered is greater than or equal to one.

**Action**

Use NEXT to display Field Menus until the Field Menu for QTY is displayed. Press FIELD TOGGLE. Type GE 1 in the lower menu, as shown in Figure 4-5. Press ENTER.

Note that the data type of the QTY field was set to DIG in a prior exercise. (Section 3). It could be any other numeric type, but not a CHAR type or one of the date types.

**Topic: Statement Syntax**

Now that you have entered some edit statements and looked at examples of others, you should be getting familiar with their format. Basically, there is the statement name followed by some sort of value or pattern. This basic format of the FORMSPEC edit statements is shown in the syntax diagram below. Notice that you also have the option of specifying a custom error message (*errmsg*) with an edit statement. Before you finish entering the edit statements used in this exercise, read the following discussion of the value, pattern, and error message parameters used with FORMSPEC edit statements.

## Syntax of the FORMSPEC Edit Statements

MINLEN *value* ["*errmsg* "]

{GT}

{LT}

{GE}

{LE} *value* ["*errmsg* "]

{EQ}

{NE}

{IN } {*value* [,*value* ]... }

{NIN} {*lowvalue:highvalue* [,*lowvalue:highvalue* ]... } ["*errmsg* "]

{*value* [,*lowvalue:highvalue* ]... }

MATCH *pattern* ["*errmsg* "]

### The Value Parameter

Values used in comparisons may be constants, field names, or expressions.

Examples of these are:

*constants* Includes values such as "JONES", 123.56, !9/1/86!, \$TODAY, \$LENGTH or \$EMPTY

*fieldnames* such as ORDNUM, QTY or NAME

*expressions* such as 5+10, QTY\*UPRICE or 6%TPRICE

**Constants.** Constants are divided into categories corresponding to the three main data types: character, numeric, and date. Character type constants must be enclosed within quotation marks and date constants between exclamation points. In addition, there are four system-defined constants:

\$TODAY means today's date and is a *date* type constant, which means the DType must be MDY, DMY, or YMD.

\$LENGTH means the current field length and is a *numeric* constant, which means the DType must be one of the numeric data types, DIG, NUM, or IMP.

\$EMPTY means a null value and can be any data type.

\$STATE means a two-character abbreviation (upper or lowercase) of the state code for one of the 50 states or 5 territories of the United States. It is a *character* type constant. The state codes are contained in a special table provided by VPLUS/V and are listed in Appendix F of the *VPLUS/V Reference Manual*.

The system-defined constants must be compatible with the data type of the field to be edited.

**Expressions.** Expressions can be formed only with *numeric* data types, and can only use the operators shown in Table 4-3.

**Table 4-3. Operators**

+ - % * /	addition subtraction percent multiplication division
-----------------------	--

**The Pattern Parameter**

The MATCH statement pattern allows you to group characters and to repeat single or grouped characters. Grouping is indicated by pairs of brackets or braces; repetition by following the character or grouped characters by a plus (+) or asterisk (\*) character. The pattern control characters are shown in Table 4-4. The choice of symbols depends on whether a character or group of characters can be omitted or not. However, leading and trailing blanks are not allowed in a MATCH pattern.

You can also specify that the pattern include a specific character such as the hyphen shown in the example above. If you want to include any of the pattern control characters as a specific character, you must precede it by an exclamation point (!). For example:

- MATCH !a???c      Value must start with the letter a followed by any three characters, and then the letter c.
- MATCH a???c      Value must start with any upper or lowercase letter followed by any three characters and then the letter lowercase c.

**Table 4-4. Pattern Control Characters**

Character	Meaning
a	upper or lowercase letter of alphabet (A-Z,a-z)
u	uppercase letter of alphabet (A-Z)
l	lowercase letter of alphabet (a-z)
b	blank (must not be specified as first or last character of pattern)
d	any digit (0-9)
?	any character

## The Error Message Parameter

A custom message is any series of up to 80 characters enclosed within quotation marks. The message should follow the field edit statement, either on the same line or a following line, and must precede any subsequent processing statements.

```
FORMSPEC v.uu.ff Field Menu                                FORM: ORDER
  Name [name ]
  Address [address ]
  City [city ] State [st]
  ^
  Num [5 ] Len [2 ] Name [ST ] Enh [HI ] FType [0] DType [CHAR]
  Initial Value [ ]
  *** Processing Specifications ***
  IN $STATE
```



Figure 4-6. Field Menu for STATE in the ORDER Form

## Use a Table Check

### Action

Use PREV to return to the Field Menu for ST; then use FIELD TOGGLE to go to the processing specification area. Type IN \$STATE, as shown in Figure 4-6. Press ENTER.

The edit statement insures that the state abbreviation entered in the field is one of the 55 official abbreviations used by the United States Post Office. Note that this field can be omitted (the field type is still optional), but if any value is entered, it must be a valid state abbreviation.

Other checks that might be made on the STATE field are:

- |                            |   |
|----------------------------|---|
| IN "CA", "NY", "FL"        | Only 3 state codes (California, New York, or Florida) are allowed and they must be all uppercase. |
| NIN "GA", "Ga", "ga", "gA" | Do not allow any state code for Georgia.  |

```

FORMSPEC v.uu.ff Field Menu                                FORM: ORDER
Address [address ]
  City [city ] State [st]
  Zip [zip ] Telephone [phone ]
  ^
Num [6 ] Len [10 ] Name [ZIP ] Enh [HI ] FType [0] DType [CHAR]
Initial Value [
*** Processing Specifications ***
MATCH ddddd[-dddd]

```

```

PREV FORM  NEXT FORM  FIELD TOGGLE  REFRESH          PREV  NEXT  MAIN/RESUME  EXIT

```

Figure 4-7. Field Menu for ZIP in the ORDER Form

## Use Pattern Matches

Now you should be at the Field Menu for the field, ZIP.

### Action

Press FIELD TOGGLE. Type MATCH ddddd[-dddd] in the lower menu, as shown in Figure 4-7. Press ENTER.

Here you use a pattern match to make sure that any entered zip code is five digits with, optionally, four more digits with a hyphen between the fifth and sixth digit.

```

FORMSPEC v.uu.ff Field Menu                                FORM: ORDER
Address [address ]
  City [city ] State [st]
  Zip [zip ] Telephone [phone ]
  ^
Num [7 ] Len [14 ] Name [PHONE ] Enh [HI ] FType [0] DType [CHAR]
Initial Value [ ]

*** Processing Specifications ***
MATCH [(ddd)]b*ddd-dddd

```



Figure 4-8. Field Menu for PHONE in the ORDER Form

A good illustration that uses some of the pattern characters is the match pattern for a telephone number.

**Action**

Press FIELD TOGGLE. Type MATCH [(ddd)]b\*ddd-dddd in the lower menu, as shown in Figure 4-8. Press ENTER.

The brackets surrounding the area code allow it to be omitted. The asterisk (\*) after the letter b allows zero spaces or as many as you want. Table 4-5 illustrates some other examples of pattern choices.

Table 4-5. Pattern Choices

Pattern	Meaning
{A,B,!a,!b}	Required grouping - Entered character corresponding to this pattern must be A, B, a, or b.
[A,B,!a,!b]	Optional grouping - No character need be specified in this position or, if entered, it must be A, B, a, or b.
d+	Required repetition - At least one digit must be entered, and as many more digits as desired may be entered.
d*	Optional repetition - No digit need be entered in this position, or as many digits as desired may be entered.
{A,B,!a,!b}+	Any of the enclosed characters can each be repeated as many times as desired - for example: BBBBaab.
{A,B,!a,!b}*	No character need be specified, or any of the enclosed characters can be repeated.

```

FORMSPEC v.uu.ff Field Menu                                FORM: ORDER
QTY  PART NO.      DESCRIPTION                                UNIT  TOTAL
                                           PRICE  PRICE
qty.. partnum..  description..... uprice. tprice.
  ^
Num [10 ] Len [9 ] Name [PARTNUM ] Enh [HI ] FType [0] DType [CHAR]
Initial Value [
*** Processing Specifications ***
MATCH Puddd-???

```

```

PREV  NEXT  FIELD  REFRESH          PREV  NEXT  MAIN/  EXIT
FORM  FORM  TOGGLE REFRESH          FORM  RESUME

```

**Figure 4-9. Field Menu for PARTNUM in the ORDER Form**

When you press ENTER at the Field Menu for PHONE, the Field Menu for QTY, which was shown earlier in Figure 4-5, is displayed on your terminal screen.

**Action**

Press NEXT to display the Field Menu for PARTNUM.

A part number also provides a good example of how to use a pattern match.

**Action**

Press FIELD TOGGLE. Type MATCH Puddd-??? in the lower menu, as shown in Figure 4-9. Press ENTER.

A part number such as PS123-XYZ or PC999-001 is accepted by the pattern match statement shown above, but Pa222-001 or BA123-TTT would cause an error.

```

FORMSPEC v.uu.ff Field Menu                                FORM: ORDER
*****ABC MANUFACTURING*****

Order Number: [ordnum]                                     Date: [ordate ]
                                                ^
Num [1 ] Len [18 ] Name [ORDATE ] Enh [HI ] FType [R] DType [MDY ]
Initial Value [ ]
-----
*** Processing Specifications ***
EQ $TODAY "Enter today's date in Month Day Year order."

PREV FORM  NEXT FORM  FIELD TOGGLE  REFRESH          PREV  NEXT  MAIN/RESUME  EXIT

```

Figure 4-10. Field Menu for ORDATE in the ORDER Form

### CUSTOM ERROR MESSAGES

Whenever you specify a field edit, you should also provide a custom error message to be issued by ENTRY when an entered value fails the field edit. Otherwise, if a field fails the edit, a standard VPLUS/V message is issued. Since the standard messages cannot be informative for a particular edit, each field edit should have a message that tells the user exactly why the entered value is incorrect. If an entered value fails the edit, the custom message is issued instead of the standard date type message. In this exercise, you will specify a custom error message for the ORDATE field.

### Action

Return to the Field Menu for ORDATE by pressing PREV until it is displayed. Then add a custom error message to the field edit, as shown in Figure 4-10. Press ENTER.

Whenever you want to continue the message onto another line of the Field Menu, put quotation marks around the characters on each line and join the two character strings with an ampersand (&). For example:

```

GE 1 "Enter quantity greater than" &
      " or equal to 1."

```

The message is displayed by ENTRY in the window line of the form as a single line.

**Table 4-6. Summary of Current Fields in the ORDER Form**

Field Num	Name	FType	DType	Processing Specifications	Initial Value
8	ORDNUM	R	CHAR	--	100000
1	ORDATE	R	MDY	EQ \$TODAY	--
2	NAME	O	CHAR	--	--
3	ADDRESS	O	CHAR	--	--
4	CITY	O	CHAR	--	--
5	ST	O	CHAR	IN \$STATE	--
6	ZIP	O	CHAR	MATCH ddddd[-dddd]	--
7	PHONE	O	CHAR	MATCH [(ddd(]b*ddd-dddd	--
9	QTY	R	DIG	GE 1	--
10	PARTNUM	R	CHAR	MATCH Puddd-???	--
11	DESCRIPTION	O	CHAR	-----	--
12	UPRICE	R	NUM2		--
13	TPRICE	R	NUM2		--

The current fields in the ORDER form should now have the simple and full edits as shown in Table 4-6. In addition, the ORDATE field has a custom error message. Actually, every field edit should have a message associated with it, which provides more helpful advice on how to correct the error than the generic message can provide.

### Action

Proceed through the Field Menus adding custom messages to each field edit statement. Table 4-7 provides suggested messages for each field edit.

**Table 4-7. Summary of Custom Error Messages**

Field Name	Edit Statement	Custom Message
ORDATE	EQ \$TODAY	Enter today's date in Month Day Year order.
ST	IN \$STATE	Enter the correct 2-character state code.
ZIP	MATCH dddd[-dddd]	Enter ZIP code, must be either 5 digits or 9 digits and a hyphen.
PHONE	MATCH [(ddd)]b*ddd-ddd	Enter phone number; any area code must be in parentheses; e.g., (408) 739-8809.
QTY	GE 1	Enter quantity equal to or greater than 1.
PARTNUM	MATCH Pudd-???	First letter must be P, then letter A-Z, 3 digits, hyphen, 3 characters.

```

FORMSPEC v.uu.ff Main Menu                                FORMS FILE: ORDFORM1
[X] Enter Selection
A--Add a form
S--Add a Save field
T--Terminal/Language Selection Menu
G--Go to GLOBALS Menu, OR Go to form [          ] field [          ]
L--List Forms File, OR List form ... [          ]
D--Delete Save field ..... [          ]
   Form ..... [          ]
C--Copy new form name ..... [          ]
   from form ..... [          ]
   from Forms File (opt) ..... [          ]
X--Compile Forms File
   Optional: Fast Forms File [          ]
R--Relate child form [          ] to parent form [          ]

PREV  NEXT  REFRESH  PREV  NEXT  MAIN/  EXIT
FORM  FORM  FORM      FORM  FORM  RESUME FORM

```

**Figure 4-11. The Main Menu, Selecting Compile**

Once you have entered all the custom error messages, you are ready to compile the forms file.

**Action**

Press MAIN/RESUME to display the Main Menu. Type an X in the selection box to compile the forms file, as shown in Figure 4-11. Press ENTER.

Unless the file is compiled, none of the new edits or error messages will be executed when the file is used with ENTRY.

```
:RUN ENTRY.PUB.SYS  
HP32209v.uu.ff ENTRY (C) HEWLETT-PACKARD CO. 1986  
ENTER FORMS FILE NAME AND PRESS RETURN: ORDFORM1  
ENTER BATCH FILE NAME AND PRESS RETURN: NEWDATA
```

```
-  
*****ABC MANUFACTURING*****  
Order Number: [100000] Date: [ ]  
  
SHIP TO:  
Name [ ]  
Address [ ]  
City [ ] State [ ]  
Zip [ ] Telephone [ ]  
  
QTY PART NO. DESCRIPTION UNIT PRICE TOTAL PRICE  
[ ] [ ] [ ] [ ] [ ]  
  
ENTRY v.uu.ff Batch Record #1 Mode: Collect  
HEAD | PRINT | REFRESH | NEXT | BROWSE | EXIT  
FORM | | | FORM | |
```

Figure 4-12. Run ENTRY; The ORDER Form, Batch Record No. 1

### USE FULL FIELD EDITS WITH ENTRY

#### Action

To enter data into the edited field, run ENTRY, as shown in Figure 4-12. When ENTRY asks for the forms file name, enter the existing file name. Then, when ENTRY requests the batch file name, enter the name of a new batch file. Use the name NEWDATA.

ENTRY then displays the first (and only) form in the file ORDFORM1. Although the form, shown in Figure 4-12, looks identical to the ORDER form displayed in Section 3, there are now more stringent edits and custom error messages on six of the fields.

```

-
*****ABC MANUFACTURING*****
Order Number: [100001] Date: [30 SEP 1986 ]

SHIP TO:
Name [MARION ENTERPRISES ]
Address [3300 Lasalle Avenue ]
City [New York ] State [NU]
Zip [8314 ] Telephone [213 632-5500 ]

QTY PART NO. DESCRIPTION UNIT TOTAL
PRICE PRICE
0 pa555-00a 10.00 100.00

Enter today's date in Month Day Year order.
HEAD PRINT REFRESH NEXT BROWSE EXIT
FORM FORM

```

Figure 4-13. The ORDER Form, Displaying Errors

**TEST THE FIELD EDITS**

In order to test the field edits added to the ORDER form, enter incorrect data in each of the edited fields.

**Action**

Type in the data shown in Figure 4-13 and press ENTER.

All six of the fully edited fields are in error. Accordingly, ENTRY highlights and underlines all six fields and displays the custom error message for the first field in the window line. If each error is corrected singly, the message associated with each edit is displayed. Try it.

**Action**

Correct the first error and press ENTER to display the custom error message for the next field with incorrect data. Continue until all data is correct and all six messages have been displayed.

The messages displayed should correspond exactly to the custom messages you specified for these fields with FORMSPEC. (Refer back to Table 4-7 for a list of these messages.) Once all the errors are corrected, the data is written to Record #1 of the new batch file, NEWDATA, and ENTRY displays the form with initial values only.

\*\*\*\*\*ABC MANUFACTURING\*\*\*\*\*

Order Number: [100002]

Date: [SEP 30 1986]

SHIP TO:

Name [SAM'S BICYCLES]  
Address [315 Lark Lane]  
City [Sunnyvale] State [Ca]  
Zip [95067] Telephone [(408) 333-4567]

QTY	PART NO.	DESCRIPTION	UNIT PRICE	TOTAL PRICE
10	PJ123-003	Pedal	15.75	157.50

ENTRY v.uu.ff Batch Record #2 Mode: Browse  
FIRST REC | DELETE REC | PRINT | REFRESH | PREV REC | NEXT REC | COLLECT | EXIT

Figure 4-14. The ORDER Form, Batch Record No. 2

#### USE REFRESH IN COLLECT MODE

Suppose you have typed most or all of the data in a form when you realize that the data is all wrong. In this case, rather than clearing the data by hand or typing over it, it is simpler to reset the form to its initial values by pressing REFRESH.

#### Action

Type some of the data into the form for Record #2, such as the example shown in Figure 4-14. Now, press REFRESH.

This causes the entire terminal screen to be cleared. Then the form is painted on the screen again with no data except initial values in any of the fields. The record number is still Record #2.

#### Action

Type the data exactly as shown in Figure 4-14. Press ENTER.

:RUN ENTRY.PUB.SYS

HP32209v.uu.ff ENTRY (C) HEWLETT-PACKARD CO. 1986

ENTER FORMS FILE NAME AND PRESS RETURN: ORDFORM1

ENTER BATCH FILE NAME AND PRESS RETURN: NEWDATA

```

-
*****ABC MANUFACTURING*****
Order Number: [100000]           Date: [          ]

SHIP TO:
  Name [          ]
  Address [          ]
  City [          ] State [    ]
  Zip [          ] Telephone [    ]

QTY   PART NO.   DESCRIPTION                               UNIT   TOTAL
      [          ] [          ] [          ] [PRICE] [PRICE]

ENTRY v.uu.ff           Batch Record #3           Mode: Collect
HEAD |          | PRINT | REFRESH |          | NEXT | BROWSE | EXIT
FORM |          |     |     |          | FORM |     |     |

```

Figure 4-15. Run ENTRY; The ORDER Form, Batch Record No. 3

#### REVIEW AND CORRECT ENTERED DATA

After entering data in a set of records, it is a good idea to proof the data. As you learned in Section 3, this can be accomplished with Browse mode.

#### Action

Press BROWSE to go into Browse/Modify mode and then press FIRST REC to view the first record in the file.

You can then proceed sequentially through the entire file, checking the data entered in each record, by pressing NEXT REC. Currently, there are only two records in the file, NEWDATA. Note that you do not have to view the data in the same session in which it was entered. For instance, if you have exited from ENTRY, you only have to request the same batch file when you next run ENTRY and then immediately press BROWSE to view the data.

#### Action

Press EXIT, then issue the run command for ENTRY, as shown in Figure 4-15. When ENTRY asks for the batch file name, type the file name NEWDATA.

ENTRY then displays the form for Record #3, also shown in Figure 4-15.

Record #3 is the next available data record in the NEWDATA file, since you have already entered the first two records.

**Action**

Press BROWSE to go to Browse mode.

\*\*\*\*\*ABC MANUFACTURING\*\*\*\*\*

Order Number: [100001]

Date: [SEP 30 1986]

SHIP TO:

Name [MARION ENTERPRISES]  
Address [3300 Lasalle Avenue]  
City [New York] State [NY]  
Zip [10034] Telephone [(213) 632-5500]

QTY	PART NO.	DESCRIPTION	UNIT PRICE	TOTAL PRICE
10	PA555-00A		10.00	100.00

ENTRY v.uu.ff Batch Record #1 Mode: Browse  
FIRST REC | DELETE REC | PRINT | REFRESH | PREV REC | NEXT REC | COLLECT | EXIT

Figure 4-16. The ORDER Form, Batch Record No. 1

When you press BROWSE, the last record entered is displayed, which in this exercise is Record #2 (shown earlier in Figure 4-16).

**Action**

Press FIRST REC to look at Record #1, as shown in Figure 4-16.

If you find any errors, correct them by typing over the data and then pressing ENTER. The corrected data is written to the batch record, overwriting the previous data. If you do not find errors, simply press NEXT REC to display the next record.

\*\*\*\*\*ABC MANUFACTURING\*\*\*\*\*

Order Number: [100002]

Date: [SEP 30 1986]

SHIP TO:

Name [SAM'S BICYCLES]  
 Address [315 Lark Lane]  
 City [Sunnyvale] State [Ca]  
 Zip [95067] Telephone [(408) 333-4567]

QTY	PART NO.	DESCRIPTION	UNIT PRICE	TOTAL PRICE
10	PJ123-003	Pedal	15.75	157.50

```

ENTRY v.uu.ff          Batch Record #2          Mode: Browse
FIRST  DELETE  PRINT  REFRESH  PREV  NEXT  COLLECT  EXIT
REC    REC
    
```

Figure 4-17. The ORDER Form, Batch Record No. 2

If you copied the data from Figure 4-14 exactly, then the word "Bicycles" is misspelled in Record #2.

**Action**

Press NEXT REC to get to Record #2. Press TAB to position the cursor at the Name field. Correct the error by typing BICYCLES over the word BYCLES. Press ENTER to write the corrected data to the batch file.

Another method of correcting the error is to use Insert char to insert the missing letters I; Insert char is turned off when you press ENTER to record the correction.

When you press ENTER, the corrected record is written to the batch file as Record #2, overwriting the original, erroneous version. If there were more records in this batch file, ENTRY would then display Record #3, still in Browse mode. However, since there are only two records in NEWDATA, ENTRY displays the message that there are no more records.

Despite full field editing, you may find many such typing errors in a batch file. Editing cannot check everything--and the type of data entered in a field of type CHAR is not checked at all.

```

-
*****ABC MANUFACTURING*****
Order Number: [100003]           Date: [SEP 30 1986 ]

SHIP TO:
Name [FORTUNE HARDWARE ]
Address [2500 Fair Oak Ave. ]
City [Mountain View ] State [Ca]
Zip [95056 ] Telephone [(408) 786-1367]

QTY   PART NO.   DESCRIPTION                               UNIT   TOTAL
      PRICE     PRICE
4     PY368-102  Hinge                                       2.50   10.00

ENTRY v.uu.ff           Batch Record #3           Mode: Browse
FIRST  DELETE  PRINT  REFRESH  PREV  NEXT  COLLECT  EXIT
REC    REC

```

Figure 4-18. The ORDER Form, Batch Record No. 3

**USE REFRESH IN BROWSE MODE**

In Browse mode, REFRESH can be used in the same manner as in Collect mode, however, when pressed in Browse mode, REFRESH does not clear all the data; it simply clears any changes typed over the original data.

**Action**

Return to Collect mode by pressing COLLECT. Type data into Record #3, as shown in Figure 4-18. Press ENTER. Now press BROWSE to verify the data entered in Record #3.

In Record #3, the original quantity of the ordered item is shown as 4. You decide that this value should be 6.

**Action**

Change the order quantity to 6 and change the Total Price field to 15.00 to correspond. Do not press ENTER yet.

Now, looking over the order one final time, you realize that the quantity really should be four. Rather than changing the quantity back to 4 and recalculating the total price, you can press REFRESH before pressing ENTER to display the original version of Record #3.

**Action**

Press REFRESH. Compare the data on your screen with Figure 4-18.

After you press REFRESH, the form contains the original values. Since you have not changed Record #3, it is not necessary to press ENTER. You have completed this session.

**Action**

Press EXIT.

## Chapter 5 ADD NEW FORM AND SPECIFY CALCULATED VALUES

Forms file ORDFORM1 now contains one form with field editing specified for data entered on that form. In this section, you add a new form to the file. The new form contains fields in which data is calculated and then displayed at the terminal without user intervention, and fields in which user-entered data is formatted. The features covered in this section are:

### Using FORMSPEC

- \* Add new form (Main Menu) display-only
- \* Establish forms sequence (Form Menu)
- \* Use display-only fields (Field Menu)
- \* Assign values with SET TO statement
- \* Use calculates values
- \* Do conditional calculations with IF statement
- \* Format data by moving it
- \* Use formatting statements (STRIP, JUSTIFY)

### Using ENTRY

- \* Enter data in forms with fields
- \* Show forms sequence
- \* review calculated values
- \* Modify calculated values

---

```
| :RUN FORMSPEC.PUB.SYS |
```

---

FORMSPEC v.uu.ff Forms File Menu

Forms File Name [ORDFORM1]

-

PREV FORM	NEXT FORM		REFRESH	PREV	NEXT	MAIN/ RESUME	EXIT
--------------	--------------	--	---------	------	------	-----------------	------

Figure 5-1. Forms File Menu

## Add New Form

### Action

Run FORMSPEC and specify the forms file ORDFORM1 when the Forms File Menu is displayed, as shown in Figure 5-1. Press ENTER.

```

[A] Enter Selection
Forms file is 1% full
A--Add a form
S--Add a Save field
T--Terminal/Language Selection Menu
G--Go to GLOBALS Menu, OR Go to form [ ] field [ ]
L--List Forms File, OR List form ... [ ]
D--Delete Save field ..... [ ]
   Form ..... [ ]
C--Copy new form name ..... [ ]
   from form ..... [ ]
   from Forms File (opt) ..... [ ]
X--Compile Forms File
   Optional: Fast Forms File [ ]
R--Relate child form [ ] to parent form [ ]

```

PREV FORM	NEXT FORM		REFRESH		PREV	NEXT	MAIN/ RESUME	EXIT
--------------	--------------	--	---------	--	------	------	-----------------	------

Figure 5-2. Main Menu

SELECT ADD OPTION

Action

When ENTER is pressed for the Forms File Menu, the Main Menu is issued. Enter A in the selection box to add a form, as shown in Figure 5-2. Press ENTER.

```

Form Name [TOTALS ]
Repeat Option [N]
           N--No Repeat
           A--Repeat, appending
           R--Repeat, overlaying

Next Form [C] Name [$HEAD ]
           C--Clear before Next Form
           A--Append Next Form
           F--Freeze, then append Next Form

Function Key Label [ ]
                 Y--Define form level function key labels.

Reproduced from [ ] (opt)

Comments [Totals for order entered on form ORDER ]

PREV  NEXT  REFRESH  PREV  NEXT  MAIN/  EXIT
FORM  FORM  FORM      FORM  FORM  RESUME

```

**Figure 5-3. Form Menu for the TOTALS Form**

When you press ENTER, a new Form Menu is displayed.

### Action

Name the new form TOTALS by typing TOTALS in the Form Name field, as shown in Figure 5-3. The other default values need not be changed. Press ENTER.

You specify the Next Form option such that this form follows the first form (ORDER) and, after the totals are entered, the first form is repeated. You could have changed the Next Form Name to ORDER to repeat the first form. However, with only two forms in the forms file, you left the default value of \$HEAD as the Next Form Name for the TOTALS form, which will achieve the same result.

```

Form Name [ORDER ]
Repeat Option [N]
           N--No Repeat
           A--Repeat, appending
           R--Repeat, overlaying

Next Form [C] Name [TOTALS ]
           C--Clear before Next Form
           A--Append Next Form
           F--Freeze, then append Next Form

Function Key Label [ ]
                 Y--Define form level function key labels.

Reproduced from [ ] (opt)

Comments [This is the first of a 2-part order form ]

PREV FORM  NEXT FORM  REFRESH  PREV  NEXT  MAIN/RESUME  EXIT

```

**Figure 5-4. Form Menu for the ORDER Form**

When you press ENTER to record the TOTALS form, a blank screen is displayed on which to design layout for the TOTALS form. Before you design the new form layout, let's return to the Form Menu for ORDER.

### Action

Press PREV FORM twice. On the Form Menu for ORDER, change the Next Form Name from \$HEAD to TOTALS and supply a new comment, as shown in Figure 5-4. Press ENTER.

As a result of the new Next Form Name, when data has been entered on the first form, ORDER, the next form issued by ENTRY is the TOTALS form.

```

-
*****ABC MANUFACTURING*****

Order Number: [ordnum]                               Date: [ordate      ]

                                           Total Net Price: [totnet  ]

Discount % [dp]                                     Discount: [discnt  ]

                                           Subtotal: [sub     ]

Tax Rate [tr  ]                                     Tax: [tax         ]

                                           Shipping: [ship    ]

                                           TOTAL: [total     ]

Customer Name [name                               ]

PREV FORM  NEXT FORM  REFRESH  PREV  NEXT  MAIN/RESUME  EXIT

```

Figure 5-5. Form Layout for TOTALS

### DESIGN THE FORM LAYOUT

Now that the forms sequence for the two forms has been established, you can design the layout for the new TOTALS form.

#### Action

First, press NEXT FORM to go to the Form Menu for TOTALS (Figure 5-3), then, press NEXT to go to the Form Layout Menu, which is a blank screen with a message that appears briefly in the upper left corner. Design the layout for the TOTALS form as shown in Figure 5-5 and described in Table 5-1. Press ENTER to record the completed layout.

Table 5-1. Defining the Fields for the TOTALS Form

Field	Tag	Line No.	Length
Order Number	ordnum	5	6
Date	ordate	5	18
Total Net Price	totnet	8	10
Discount %	dp	10	2
Discount	discnt	10	10
Subtotal	sub	12	10

Tax Rate	tr	14	4
Tax	tax	14	10
Shipping	ship	16	10
Total	total	19	10
Customer Name	name	22	42

---

**TOPIC: DISPLAY-ONLY FIELDS**

Now that the new form, TOTALS, is defined, you establish display-only fields in both forms. A display-only field is one in which data is generated by VPLUS/V or the application rather than entered by the user. Display-only fields are protected so the user cannot enter or modify data in these fields.

The first field to be made display-only is the date. Since the date in the ORDER form must always be today's date (edit statement EQ \$TODAY), you prevent potential user error by having VPLUS/V calculate this date. Similarly, you specify in FORMSPEC that the date in the TOTALS form is to be calculated by VPLUS/V.

You also change to display-only the fields that the user should not be able to change because the totals are calculated by ENTRY. In the first form, ORDER, there is only one totals field, the TPRICE field. Instead of having the user calculate and then enter this value, VPLUS/V can calculate the value by multiplying the user-entered values QTY and UPRICE. In the second form, TOTALS, there are several fields whose values can be calculated from other values in the form. These fields are: DISCOUNT, TAX, and TOTAL. The obvious advantage of having VPLUS/V calculate and enter values in such fields is that arithmetic errors are avoided, and the data is entered more quickly.

To change a field to display-only, locate the Field Menu for the field and then change the field type (FType) to D for display-only. This specification prevents the user from entering data in the field, and allows the field to be assigned a value by VPLUS/V or the application. This value can be assigned directly, be calculated, or be moved from another field or from a save field. This section discusses directly assigned and calculated values. (Refer to Section 6 for values transferred from other fields, or through save fields, from other forms).

The SET statement can be specified in any of the three formats:

```

SET destination TO source
SET destination
SET [currentfield] TO source

```

This statement basically moves a value from a *source* that can be a

field,  
or a constant value, or a calculated value to a *destination* field. If the *destination* or *source* is not specified, the value of the current field, *currentfield*, is assumed. The *destination* is always a field, either a specified field name, the current field, or a save field name. The *source* can be a field, a constant, an expression, or an index retrieve operand. Note that, unlike the edit statements, the various SET statements cannot have a custom error message.

**Constants** in SET statements are specified just like constants in edit statements; that is, a character constant is surrounded by quotes ("JONES"), a date constant by exclamation points (!February 20, 1986!), and a numeric constant has no delimiters (12.35). In every case, the constant must be the same data type as the field to which it is being moved.

**Expressions** are always numeric. Expressions allow you to calculate values based on the value of numeric fields and/or constants. Expressions are very useful to calculate values and assign the result to a display-only field.

**Index retrieve** operands consist of an index and a list of values, one of which is selected based on the value of the index.

### Action

For a full discussion of the SET statement, refer to Section 4 of the *VPLUS/V Reference Manual*.

The exercise used in this guide does not need index retrieve operands, so their use is not fully depicted here. However, since they can be useful in many applications, you should be aware of them. One particular application is when you design a form to act as a menu, from which the user selects the next form. In such a case, an index retrieve operand used with the CHANGE statement (see Section 7) can be very useful. To illustrate,

```
CHANGE NFORM TO X OF FORMA, FORMB, FORMC, FORMD
```

Depending on the value of X, which can be assigned a numeric value as a result of user entry on a menu form, the next form displayed is one of the forms named in the list.

```

FORMSPEC v.uu.ff Field Menu                                FORM: ORDER
*****ABC MANUFACTURING*****

Order Number: [ordnum]                                     Date: [ordate   ]
                                                ^
Num [1 ] Len [18 ] Name [ORDATE ] Enh [HI ] FType [D] DType [MDY ]
Initial Value [ ]
-----
*** Processing Specifications ***
SET ORDATE TO $TODAY

```



**Figure 5-6. Field Menu for ORDATE in the ORDER Form**

**DEFINE FIELDS FOR FORM ORDER**

Let's use the SET TO statement to assign specified values to display-only fields.

**Action**

Locate the Field Menu for ORDATE in the ORDER form.

One method is to press PREV FORM twice and then press NEXT until the Field Menu for ORDATE is displayed. Another method is to use the Main Menu to go to the specified field. Which method did you use?

Now, to assign today's date to the ORDATE field, use the same system-defined constant \$TODAY that is used to edit dates (See Section 4). A value is assigned to the current field with the SET TO statement.

**Action**

In order to enter this statement, press FIELD TOGGLE to move the cursor to the Processing Specifications area of the Field Menu for ORDATE. Delete the previous statement and type in the new statement SET ORDATE TO \$TODAY as shown in Figure 5-6; press ENTER.

The SET edit statement is used in order to assign the value of today's date to the ORDATE field. Note that the date is in the form *mm/dd/yy*. The order, MDY, is determined by the data type of the field. (In order

to assign a date to a field, the data type (DType) must be MDY, DMY, or YMD.) The Field Menu for ORDATE in the ORDER form now appears as shown in Figure 5-6.

```

FORMSPEC v.uu.ff Field Menu                                FORM: ORDER
QTY  PART NO.      DESCRIPTION                                UNIT  TOTAL
                                           PRICE  PRICE
qty.. partnum..  description.....  uprice. tprice.
                                           ^
Num [15 ] Len [7  ] Name [TPRICE ] Enh [HI ] FType [D] DType [NUM2]
Initial Value [
*** Processing Specifications ***
SET TPRICE TO QTY * UPRICE

```

```

PREV  NEXT  FIELD  REFRESH
FORM  FORM  TOGGLE

PREV  NEXT  MAIN/  EXIT
FORM  FORM  RESUME

```

Figure 5-7. Field Menu for TPRICE in the ORDER Form

### CALCULATED VALUES

In addition to assigning specified values to display-only fields, the SET TO statement is also used to assign *calculated* values to display-only fields. In this case, the data type of all the fields involved in the calculations must be numeric. That is, their data type must be DIG, NUM[n], or IMPn. To illustrate, let's calculate the total cost of the part ordered through the ORDER form.

#### Action

First, locate the Field Menu for TPRICE in the ORDER form.

Since the TPRICE field is the last of 13 fields in the ORDER form, the simplest method is to press NEXT FORM to go to the Form Menu for TOTALS, and then press PREV to go back to the last Field Menu of the previous form, ORDER. Or, you can go to the Main Menu, select G, the form name (TOTALS) and the field name (TPRICE) to go to the particular field.

#### Action

When the Field Menu for TPRICE is displayed, change the field type to D.

Press FIELD TOGGLE. Type the SET TO statement shown in Figure 5-7 in the processing specification area. Press ENTER.

The SET TO statement calculates the value of TPRICE from UPRICE and QTY. The data type for TPRICE is already NUM2 so it need not be changed. Note that the data types of QTY and UPRICE are already numeric; QTY is type DIG and UPRICE is NUM2.

As a result of this specification, the user need only enter values in the QTY field and the UPRICE field. When ENTER is pressed, ENTRY calculates the total price for the TPRICE field and transfers it to the batch record with the other data from this form. The user does not see the calculated value unless using Browse mode to review the data.

Any field used in a calculation must have a value when the calculation is performed. If it has no value, then the calculation is skipped. In this exercise, the values of the QTY and UPRICE fields are both required (FType of R) so that the calculation of TPRICE is always performed.

```
FORMSPEC v.uu.ff Field Menu                                FORM: TOTALS
*****ABC MANUFACTURING*****

Order Number: [ordnum]                                     Date: [ordate      ]
                                                                    ^
Num [2  ] Len [18  ] Name [ORDATE      ] Enh [HI  ] FType [D] DType [MDY ]
Initial Value [
-----
*** Processing Specifications ***
SET ORDATE TO $TODAY

PREV FORM  NEXT FORM  FIELD TOGGLE  REFRESH          PREV  NEXT  MAIN/RESUME  EXIT
```

Figure 5-8. Field Menu for ORDATE in the TOTALS Form

**DEFINE FIELDS FOR THE TOTALS FORM**

Now that the display-only fields in the ORDER form are defined, you need to define the display-only fields for the TOTALS form in a similar manner.

**Action**

Refer to the form layout for TOTALS, which was shown earlier in Figure 5-5.

Since the TOTALS form, as its name implies, contains the totals for the order, the fields contain mainly dollar values and percentages. Thus, all the fields are numeric, except the order number, the date, and the customer name fields. Another feature of this form is that very few items require user entry. Most fields that are not display-only can be assigned initial values. For example, the discount percentage and the tax rate, which change infrequently, will be assigned initial values. The discount is calculated from the discount percentage, the tax from the tax rate, and the total price is calculated from these and other dollar values. In fact, the only values the user need enter are the order number and the net price; shipping costs and customer name are optional.

**Action**

Locate the Field Menu for ORDATE in the TOTALS Form. As was done for the ORDATE field in the ORDER form, first change the field type to D for display-only and the data type to MDY. Then press FIELD TOGGLE to go to the Processing Specifications area of the menu. Now type in the statement: SET ORDATE TO \$TODAY, as shown in Figure 5-8. Press ENTER.

Do not supply a custom message since such messages are illegal with the SET statement.

```
FORMSPEC v.uu.ff Field Menu                                FORM: TOTALS

Total Net Price: [totnet  ]
                  ^
Num [3 ] Len [10 ] Name [TOTNET ] Enh [HI ] FType [R] DType [NUM2]
Initial Value [ ]

*** Processing Specifications ***

PREV  NEXT  FIELD  REFRESH
FORM  FORM  TOGGLE

PREV  NEXT  MAIN/  EXIT
FORM  FORM  RESUME
```

Figure 5-9. Field Menu for TOTNET in the TOTALS Form

```

FORMSPEC v.uu.ff Field Menu                                FORM: TOTALS
Total Net Price: [totnet  ]
Discount % [dp]                                           Discount: [disct  ]
^
Num [4 ] Len [2 ] Name [DISCNT_PERCNT ] Enh [HI ] FType [0] DType [NUM0]
Initial Value [10 ]
*** Processing Specifications ***

PREV  NEXT  FIELD  REFRESH
FORM  FORM  TOGGLE

PREV  NEXT  MAIN/  EXIT
FORM  FORM  RESUME

```

**Figure 5-10. Field Menu for DISCNT\_PERCNT in the TOTALS Form**

When you press ENTER at the Field Menu for ORDATE, the next Field Menu displayed is for the TOTNET field, which contains the total net price of the current order. This field is entered by the user; it should equal the total order price of the TPRICE field, which was calculated on the previous form, ORDER. Note that this field must be numeric since it is used in later calculations.

#### **Action**

At the Field Menu for TOTNET, as shown in Figure 5-9, change the field type to R for required, and the data type to NUM2. Press ENTER.

When you press ENTER, the next Field Menu is displayed, which describes the discount percentage field DP.

#### **Action**

At the Field Menu for DP, as shown in Figure 5-10, change the field name to DISCNT\_PERCNT (a more descriptive name than DP); leave the field type as 0 for optional; change the data type to NUM0; and assign an initial value of 10 for a discount of 10 percent. Press ENTER.

In the next section, you learn how to transfer values between forms using Save Fields. This technique allows the order number, the total net price, and the customer name to be transferred from the ORDER form to the TOTALS form.

## TOPIC: IF STATEMENT

The *condition* in an IF statement that determines which statements are executed is always a simple edit statement. It may be preceded by a field name to indicate the field to which the edit applies. If there is no field name, the edit statement applies to the current field.

```
IF [value ] editstatement THEN [statement ]... [statement ]...  
  
[ELSE [statement ]...]  
[[statement ]... ]
```

When the condition is executed, rather than causing the field to fail if the edit is not true, it causes the ELSE part of the IF statement to be executed. If there is no ELSE, then nothing happens. Should the edit statement be true, then the statement (or statements) following THEN are executed.

The statements following THEN or ELSE can be on the same line with these key words or on the following lines. (There is no limit to the number of statements in either part.) When statements are on lines following THEN or ELSE, they *must* be indented. The word ELSE must be in the same column (indented the same number of spaces) as its preceding IF. These conventions allow you to nest IF statements.

### Action

For a full discussion of the IF statement, refer to Section 4 of the *VPLUS/V Reference Manual*.

```

FORMSPEC v.uu.ff Field Menu                                FORM: TOTALS
Total Net Price: [totnet  ]
Discount % [dp]      Discount: [discnt  ]
^
Num [5  ] Len [10  ] Name [DISCNT  ] Enh [HI  ] FType [D]  DType [NUM2]
Initial Value [
*** Processing Specifications ***

IF TOTNET GE 1000 THEN
  SET DISCNT TO DISCNT_PERCNT % TOTNET
ELSE SET DISCNT TO 0

PREV FORM  NEXT FORM  FIELD TOGGLE  REFRESH          PREV  NEXT  MAIN/RESUME  EXIT

```

Figure 5-11. Field Menu for DISCNT

### CONDITIONAL CALCULATIONS

When you press ENTER to record DISCNT\_PERCNT, the menu for the next field, DISCNT, is displayed, as shown in Figure 5-11. The DISCNT field illustrates a new statement. In most cases, a discount is granted conditionally. In this exercise, you give the customer a discount only if the total net value of the order is at least \$1,000. Such a discount can be calculated automatically using the IF statement. The IF statement causes an operation to be performed only if a specified condition is true.

#### Action

First, set the field type of DISCNT to D for display-only and the data type to NUM2. Then, press FIELD TOGGLE and type the IF statement, as shown in Figure 5-11; press ENTER.

Assuming that DISCNT\_PERCNT is 10, DISCNT is set to 10 percent of any net total equal to or greater than \$1,000. If the net total is less than \$1,000, the ELSE statement is executed to set DISCNT to 0.

```

FORMSPEC v.uu.ff Field Menu                                FORM: TOTALS
Discount % [dp]                                           Discount: [discnt  ]
                                                           Subtotal: [sub     ]
                                                           ^
Num [6  ] Len [10 ] Name [SUB                            ] Enh [HI  ] FType [D] DType [NUM2]
Initial Value [
*** Processing Specifications ***

SET SUB TO TOTNET - DISCNT

```

```

PREV FORM  NEXT FORM  FIELD TOGGLE  REFRESH          PREV  NEXT  MAIN/RESUME  EXIT

```

Figure 5-12. Field Menu for SUB in the TOTALS Form

#### MORE CALCULATED FIELDS

The next field after DISCNT is called SUB and contains the subtotal. This value is the result of subtracting any discount from the net total. The subtotal is another display-only field containing a dollar value.

#### Action

Change the field type (FType) to D and the data type (DType) to NUM2. Then press FIELD TOGGLE to enter the SET statement that calculates a subtotal, as shown in Figure 5-12. Press ENTER.

```

FORMSPEC v.uu.ff Field Menu                                FORM: TOTALS
Subtotal: [sub      ]
Tax rate [tr  ]
      ^
Num [7  ] Len [4  ] Name [TAX_RATE  ] Enh [HI  ] FType [R] DType [NUM2]
Initial Value [6.25 ]
*** Processing Specifications ***

```

```

PREV  NEXT  FIELD  REFRESH
FORM  FORM  TOGGLE

PREV  NEXT  MAIN/  EXIT
FORM  FORM  RESUME

```

**Figure 5-13. Field Menu for TAX\_RATE in the TOTALS Form**

After you press ENTER at the SUB field, the next Field Menu is the one for the tax rate (TR) field.

### Action

As shown in Figure 5-13, change the name of this field to TAX\_RATE for clarity in subsequent statements. Make it a required field, and give it the data type NUM2. Since tax rates are fairly stable, assign this field an initial value of 6.25 for a tax rate of 6 1/4%. Press ENTER.

The data type is R for required rather than D for display-only so that the user can change the initial tax rate as needed.

```

FORMSPEC v.uu.ff Field Menu                                FORM: TOTALS
Subtotal: [sub      ]
Tax rate [tr  ]                                           Tax: [tax      ]
^
Num [8  ] Len [10 ] Name [TAX      ] Enh [HI  ] FType [D]  DType [NUM2]
Initial Value [
*** Processing Specifications ***
SET TAX TO TAX_RATE % SUB

```

```

PREV FORM  NEXT FORM  FIELD TOGGLE  REFRESH          PREV  NEXT  MAIN/RESUME  EXIT

```

**Figure 5-14. Field Menu for TAX in the TOTALS Form**

The next Field Menu is for the TAX field, which is a display-only field calculated from the tax rate and the discounted total.

**Action**

As shown in Figure 5-14, change the field type to D, the data type to NUM2 to correspond to the other dollar values, and then go to the Processing Specifications to enter the SET statement to calculate the tax. Press ENTER.

Assuming the tax rate is 6.25, this statement sets the TAX field to 6 1/4 percent of the subtotal value in the SUB field. Remember that the value of SUB is the net total less the discount.

```

FORMSPEC v.uu.ff Field Menu                                FORM: TOTALS
      Tax rate [tr  ]                                     Tax: [tax  ]
                                                    Shipping: [ship  ]
                                                    ^
Num [9  ] Len [10 ] Name [SHIP  ] Enh [HI  ] FType [0] DType [NUM2]
Initial Value [.00  ]
*** Processing Specifications ***

```

```

PREV  NEXT  FIELD  REFRESH          PREV  NEXT  MAIN/  EXIT
FORM  FORM  TOGGLE  REFRESH          FORM  FORM  RESUME  FORM

```

**Figure 5-15. Field Menu for SHIP in the TOTALS Form**

The next Field Menu displayed is for the SHIP field, which is the shipping cost value. Shipping cost is a user-entered value that may or may not be needed.

**Action**

At the Field Menu for SHIP, leave the field type as 0 for optional, change the data type to NUM2 to conform to the other dollar values, and provide an initial value of zero (in the format .00 as shown in Figure 5-15).

Since the value of the SHIP field is used in a later calculation, it must have a value or the calculation would not be made.

```
                                TOTAL: [total   ]
                                ^
Num [10 ] Len [10 ] Name [TOTAL   ] Enh [HI  ] FType [D]  DType [NUM2]
Initial Value [                ]
-----
*** Processing Specifications ***

SET TOTAL TO SUB + TAX + SHIP
```



**Figure 5-16. Field Menu for TOTAL in the TOTALS Form**

The next field, TOTAL, contains the total cost of the order. The value of TOTAL is calculated by adding the tax and any shipping cost to the subtotal.

**Action**

Change the field type to D and the data type to NUM2. Then enter the SET statement in the Processing Specifications area, as shown in Figure 5-16. Press ENTER.

The last field on the TOTALS form contains the customer name originally entered on the ORDER form.

**Action**

Leave this field as is.

At this point, the user can enter a name or leave the field blank; later you will see how the name can be transferred automatically from the ORDER form.

**TOPIC: DATA FORMATTING**

Data generated by FORMSPEC as an initial value or as a display-only value is formatted automatically according to certain general rules. This

formatting depends on the data type. Table 5-1 summarizes automatic formatting. Note that whenever data is moved from one field to another (with SET TO for example) the data is formatted during the move. Unless it is an initial value, is moved, or is specifically formatted by a formatting command, user-entered data is not formatted but is stored exactly as it was entered.

As a result of automatic formatting, the date in the ORDER and TOTALS forms is always in the format *mm/dd/yy* (both dates are type MDY). The calculated display-only values are right justified with leading zeros stripped. A decimal point is inserted the correct number of positions from the right (in NUM2 fields, two positions from the right).

**Action**

For a full discussion of this topic, read "Automatic Formatting" in Section 4 of the **VPLUS/V Reference Manual**.

**Table 5-2. FORMSPEC Automatic Formatting**

Data	Formatting
CHAR	None
NUMn	Right justify value; strip leading zeros and replace with blanks; strip any plus sign; move any minus sign to position preceding first non-blank digit; strip any commas; insert decimal point as needed and fill fractional part with zeros.
NUM	Same as NUMn except that an attempt is made to fit 9 decimal places, rounding or truncating and inserting a decimal point as needed.
NUM0	Same as NUMn except that no decimal point is inserted and any fractional part is rounded.
IMPn	Same as NUMn except that any decimal point is moved.
DIG	Right justify value; strip leading zeros and replace with blanks. (Any sign, commas, or decimal point are illegal in a DIG field.)
DMY, YMD, MDY	Any date is formatted as <i>dd/dd/dd</i> where the order of the day, month, year is determined by the data type.

```

FORMSPEC v.uu.ff Field Menu                                FORM: TOTALS
      Tax rate [tr ]                                     Tax: [tax ]
                                                    Shipping: [ship ]
                                                    ^
Num [9 ] Len [10 ] Name [SHIP ] Enh [HI ] FType [0] DType [NUM2]
Initial Value [.00 ]
-----
*** Processing Specifications ***

JUSTIFY RIGHT
STRIP LEADING "0"

-

PREV  NEXT  FIELD  REFRESH
FORM  FORM  TOGGLE

PREV  NEXT  MAIN/  EXIT
FORM  FORM  RESUME

```

Figure 5-17. Field Menu for SHIP in the TOTALS Form

#### FORMATTING STATEMENTS

On the other hand, the two numeric values entered by the user in the TOTNET and SHIP fields are not formatted. You can, however, use formatting statements to align these entered values with the calculated dollar value in other fields. Specifically, to format the values of the TOTNET and SHIP fields in the TOTALS form, use the data formatting statements STRIP and JUSTIFY.

#### Action

Locate the Field Menu for SHIP in the TOTALS form. Specify the formatting statements, as shown in Figure 5-17. Press ENTER. Next, locate the TOTNET field in the TOTALS form. Give this field the same processing specifications. Press ENTER. Finally, for the sake of consistency, return to the ORDER form and format the UPRICE field in the same way as the SHIP and TOTNET fields on the TOTALS form. Press ENTER.

With these changes, all the dollar fields in both forms are formatted in the same way. Table 5-3 lists the field specifications for the ORDER form; Table 5-4 lists the field specifications for the TOTALS form.

Note that there is no way to provide a decimal point automatically for NUM[n] type fields, as is done with IMPn type fields. If a decimal point is not entered by the user, ENTRY assumes that a decimal point should follow the last digit, but does not provide the decimal point or any fractional values.

In addition to STRIP and JUSTIFY (illustrated above), FORMSPEC provides two other formatting statements: FILL and UPSHIFT. The FILL statement replaces any leading or trailing blanks in a field with any printing character you choose. The UPSHIFT statements shifts all lowercase letters to uppercase.

**Action**

Press MAIN/RESUME to return to the Main Menu. Select X to compile the forms file and press ENTER. Press EXIT when done.

FORMSPEC compiles the new forms into the forms file ORDFORM1. Until it is compiled, none of the changes will be effective when used with ENTRY.

**Table 5-3. Summary of Current Fields in the ORDER Form**

Num	Name	FType	DType	Processing Specifications	Initial Value
8	ORDNUM	R	CHAR	--	100000
1	ORDATE	D	MDY	SET ORDATE TO \$TODAY	--
2	NAME	O	CHAR	--	--
3	ADDRESS	O	CHAR	--	--
4	CITY	O	CHAR	--	--
5	ST	O	CHAR	IN \$STATE	--
6	ZIP	O	CHAR	MATCH dddd[-dddd]	--
7	PHONE	O	CHAR	MATCH [(ddd)]b*ddd-dddd	--
9	QTY	R	DIG	GE 1	--
10	PARTNUM	R	CHAR	MATCH Pudd-???	--
11	DESCRIPTION	O	CHAR	--	--
12	UPRICE	R	NUM2	JUSTIFY RIGHT STRIP LEADING "0"	--
13	TPRICE	D	NUM2	SET TPRICE TO QTY * UPRICE	--

Table 5-4. Summary of Current Fields in the TOTALS Form

Field					
Num	Name	FType	DType	Processing Specifications	Initial Value
1	ORDNUM	R	CHAR	--	100000
2	ORDATE	D	MDY	SET ORDATE TO \$TODAY	--
3	TOTNET	R	NUM2	JUSTIFY RIGHT STRIP LEADING "0"	--
4	DISCNT_PERCNT	O	NUM0	--	10
5	DISCNT	D	NUM2	IF TOTNET GE 1000 THEN SET DISCNT TO DISCNT_PERCNT % TOTNET ELSE SET DISCNT TO 0	--
6	SUB	D	NUM2	SET SUB TO TOTNET - DISCNT	--
7	TAX_RATE	R	NUM2	--	6.25
8	TAX	D	NUM2	SET TAX TO TAX_RATE % SUB	--
9	SHIP	O	NUM2	JUSTIFY RIGHT STRIP LEADING "0"	.00
10	TOTAL	D	NUM2	SET TOTAL TO SUB + TAX + SHIP	--
11	NAME	O	CHAR	--	--

:RUN ENTRY.PUB.SYS

HP32209v.uu.ff ENTRY (C) HEWLETT-PACKARD CO. 1986

ENTER FORMS FILE NAME AND PRESS RETURN: ORDFORM1

ENTER BATCH FILE NAME AND PRESS RETURN: DATA5

\*\*\*\*\*ABC MANUFACTURING\*\*\*\*\*

Order Number: [100001]

Date: [ ]

SHIP TO:

Name [MARY MCCARTHY ENTERPRISES]  
Address [10003 Haver Drive]  
City [Newport Beach] State [Ca]  
Zip [90299] Telephone [(213) 456-9900]

QTY	PART NO.	DESCRIPTION	UNIT PRICE	TOTAL PRICE
2	PA123-abc		10.50	

ENTRY v.uu.ff	Batch Record #1	Mode: Collect
HEAD FORM	PRINT REFRESH	NEXT FORM BROWSE EXIT

Figure 5-18. Run ENTRY; Record No. 1

#### DISPLAY, REVIEW, AND MODIFY CALCULATED VALUES

To enter data into the new forms, you run ENTRY,

#### Action

Issue the RUN command for ENTRY as shown. Use the forms file name ORDFORM1 and a new batch file name, DATA5.

When ENTRY displays the first form, ORDER, enter values in any unprotected fields, as shown in Figure 5-18. Press ENTER to record the data.

Because the date field is now protected, the cursor moves directly to the customer name field after you enter the new order number. Similarly, the cursor is positioned back at the order number field if you try to tab to the total price field, now a protected field.

When you press ENTER to record the data on this form, ENTRY calculates today's date and the total price and records these values in the batch record along with the values you entered. It then displays the next blank form.

Display-only values, such as the date and totals, are not shown on the screen during data collection. They do, however, appear if you browse the record.

\*\*\*\*\*ABC MANUFACTURING\*\*\*\*\*

Order Number: [100001]

Date: [09/02/86]

SHIP TO:

Name [MARY MCCARTHY ENTERPRISES ]  
Address [10003 Haver Drive ]  
City [Newport Beach ] State [Ca]  
Zip [90299 ] Telephone [(213) 456-9900]

QTY	PART NO.	DESCRIPTION	UNIT PRICE	TOTAL PRICE
2	PA123-abc		10.50	21.00

ENTRY v.uu.ff Batch Record #1 Mode: Browse  
FIRST REC | DELETE REC | PRINT | REFRESH | PREV REC | NEXT REC | COLLECT | EXIT

Figure 5-19. Record No. 1, Browse Mode

#### REVIEW CALCULATED VALUES

When you press ENTER to record the data on the ORDER form, ENTRY displays the next form, TOTALS. Before entering any data on TOTALS, let us view the data calculated by ENTRY for record #1.

#### Action

To do this, press BROWSE.

The ORDER form is displayed with all the data you entered, plus the date and the total price supplied by ENTRY, as shown in Figure 5-19.

Note that today's date is now displayed in the date field in the form *mm/dd/yy*. The total price for this order is also displayed. Make a note of this total before returning to Collect mode.

```

-
*****ABC MANUFACTURING*****
Order Number: [100001]           Date: [          ]

Total Net Price: [          ]
Discount % [10]                 Discount: [          ]
Subtotal: [          ]
Tax Rate [6.25]                 Tax: [          ]
Shipping: [.00]                 TOTAL: [          ]

Customer Name [          ]

ENTRY v.uu.ff      Batch Record #2      Mode: Collect
HEAD |          | PRINT | REFRESH |          | NEXT | BROWSE | EXIT
FORM |          |      |      |          | FORM |      |

```

Figure 5-20. Record No. 2

### Action

Press COLLECT to display the next form, TOTALS.

Four of the fields on this form have initial values as shown in Figure 5-20. For the order, you can change only the value of the order number, although any of the other initial values may be changed since they are not protected.

### Action

Leave the order number the same as the one that you entered on the ORDER form. Enter the total calculated on the ORDER form in the Total Net Price field. The cursor is then positioned at the Discount % field. On this order, leave the Discount % as 10 and the tax rate as 6.25 by pressing TAB twice. The next field you can change is the Shipping cost. Enter the value 1.25 as the shipping cost and press ENTER.

```

-
*****ABC MANUFACTURING*****
Order Number: [100001] Date: [09/02/86 ]

Total Net Price: [ 21.00]
Discount % [10] Discount: [ .00]
Subtotal: [ 21.00]
Tax Rate [6.25] Tax: [ 1.31]
Shipping: [ 1.25]
TOTAL: [ 23.56]

Customer Name [ ]

ENTRY v.uu.ff Batch Record #2 Mode: Browse
FIRST DELETE PRINT REFRESH PREV NEXT COLLECT EXIT
REC REC REC REC REC REC REC REC

```

**Figure 5-21. Record No. 2, Browse Mode**

You can examine the totals calculated for this record using Browse mode.

**Action**

Press BROWSE.

As shown in Figure 5-21, not only have the subtotal, the tax, and the final total been calculated, but all the numeric values are formatted.

**Action**

Press COLLECT to return to collect mode.

\*\*\*\*\*ABC MANUFACTURING\*\*\*\*\*

Order Number: [100002]

Date: [09/02/86]

SHIP TO:

Name [JOHNSON, SMITH, & CO.]  
Address [3269 6th Avenue South]  
City [Fairdale] State [Ca]  
Zip [95505] Telephone [(713) 231-7700]

QTY	PART NO.	DESCRIPTION	UNIT PRICE	TOTAL PRICE
10	PU111-abc		125.50	1255.00

ENTRY v.uu.ff Batch Record #3 Mode: Browse  
FIRST REC | DELETE REC | PRINT | REFRESH | PREVIOUS REC | NEXT REC | COLLECT | EXIT

Figure 5-22. Record No. 3, Browse Mode

You can now continue entering orders. On the next order, you specify a quantity and a unit price sufficient to give a discount.

**Action**

Copy the example shown in Figure 5-22. Use a quantity of 10 and a unit price of 125.50 on the ORDER form. Press ENTER.

```

-
*****ABC MANUFACTURING*****
Order Number: [100002] Date: [09/02/86 ]

Total Net Price: [ 1255.00]
Discount % [10] Discount: [ 125.50]
Subtotal: [ 1129.50]
Tax Rate [6.25] Tax: [ 70.59]
Shipping: [ .00]

TOTAL: [ 1200.09]

Customer Name [ ]

ENTRY v.uu.ff Batch Record #4 Mode: Browse
FIRST DELETE PRINT REFRESH PREV NEXT COLLECT EXIT
REC REC REC REC REC REC REC REC

```

Figure 5-23. Record No. 4, Browse Mode

**Action**

Next, enter the total net price as 1255.00 on the TOTALS form, as shown in Figure 5-23. Press ENTER.

You can assume that with an order this large, no shipping is charged, therefore you leave the shipping cost at its initial zero value.

**Action**

Browse the two records you just entered; they should appear as shown in Figure 5-22 and Figure 5-23.

```

-
*****ABC MANUFACTURING*****
Order Number: [100002]                      Date: [09/02/86 ]

Total Net Price: [ 1255.00]
Discount % [8 ]                               Discount: [ 100.40]
Subtotal: [ 1154.60]
Tax Rate [6.25]                               Tax: [ 72.16]
Shipping: [ .00]
TOTAL: [ 1226.76]

Customer Name [ ]

ENTRY v.uu.ff      Batch Record #4      Mode: Browse
FIRST  | DELETE | PRINT | REFRESH |      PREV | NEXT | COLLECT | EXIT
REC    | REC   |      |         |      REC | REC  |         |

```

Figure 5-24. Record No. 4, Browse Mode

#### MODIFY VALUES USED IN CALCULATIONS

Suppose you want to change the discount percent on this last order to 8 percent rather than 10 percent.

#### Action

To do this, remain in Browse mode, TAB to the Discount % field, and change the displayed value to 8. Press ENTER.

When you press ENTER to record the new value, the discount, the subtotal, the tax, and the final total are all recalculated. The new values are written to the existing batch record, replacing the previous values.

#### Action

When you look at the changed record, you see the new values, as shown in Figure 5-24.

Unless the record with changed values is the last record in the batch file, ENTRY does not show the new values automatically, but goes directly to the next record when you press ENTER. In this case, press PREV REC to view the changed values.

#### Action

When you are through entering and modifying data in the DATA5 batch file, you terminate ENTRY by pressing EXIT.



## Chapter 6 Using Save Fields and Understanding Phases

Values entered on one form of a forms file can be transferred to another form through the use of **save fields**, which are fields in the forms file that are independent of any particular form. As such, they can be used to hold values for transfer between forms. Save fields are defined on the Save Field Menu. This section will discuss the definition and use of Save Fields.

Processing statements can be executed before, after, or while the form is displayed. These different times to execute processing statements are called **phases** and are defined as initialization (before), field edits (while), and finish (after) phases. Phases, in general, are discussed in this section, and use of the initialization phase is illustrated.

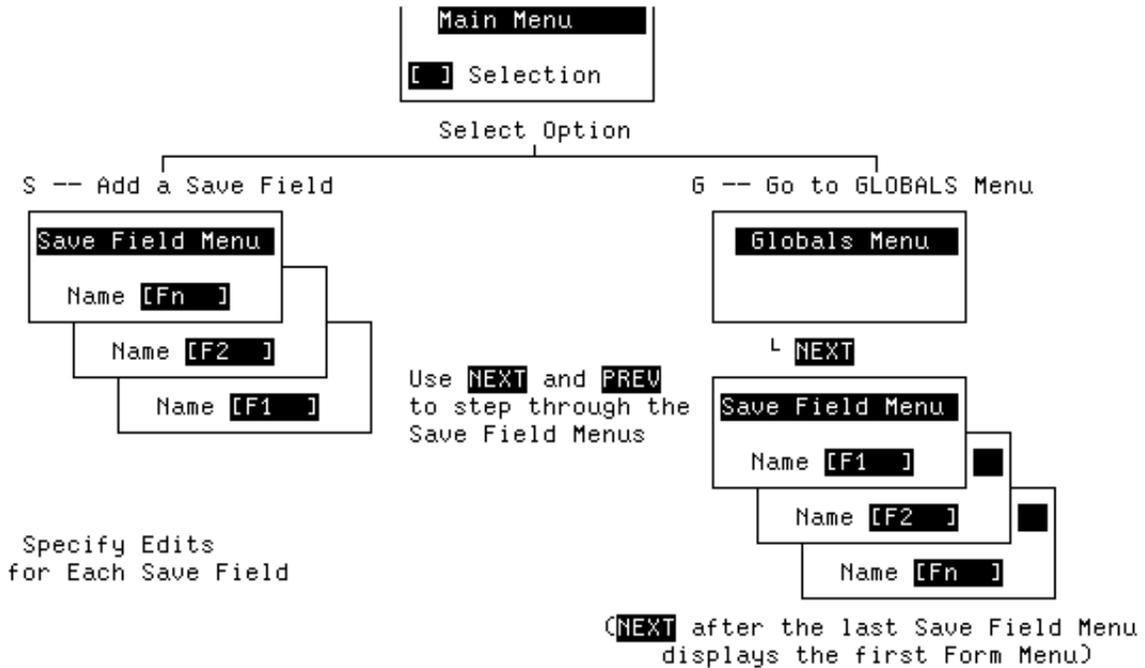
In the exercise in this section, you use three save fields to transfer the order number, total net price, and the customer name from the ORDER form to the TOTALS form. In addition, through the use of phases, you make the order number completely automatic by incrementing its value each time the ORDER form is newly displayed. Later, when the forms file is compiled with these various changes, you use ENTRY to show how values are incremented, transferred between forms, and saved even when ENTRY is terminated and restarted. The features covered in this section are:

Using FORMSPEC:

- \* Add and Modify Save Fields.
- \* Use Save Fields to assign values to fields
- \* Discuss phases in form processing.
- \* Use the initialization phase.
- \* Use Save Fields to pass values between forms.

Using ENTRY:

- \* Enter data for calculated values.
- \* Review calculated values.
- \* Terminate and resume data entry.
- \* Modify data transferred between forms.



**Figure 6-1. The Menu Sequence for Global and Save Field Menus**

As shown in Figure 6-1, there are two paths to the Save Field Menus. Remember, with either path, you must first reach the Main Menu by running FORMSPEC and requesting the forms file by name on the Forms File Menu.

Once at the Main Menu, one path is to enter S in the selection box at the Main Menu and press ENTER. A Save Field Menu will be displayed (shown later in this section) on which you can define a new save field. You define or modify save fields in the same manner as you define or modify other field or form definitions. That is, you change those values you want to modify and then press ENTER to record the changes.

The other path is a result of the fact that the Save Field Menus are positioned between the Globals Menu (discussed in Section 7) and the first Form Menu. This positioning off the main stream of form definition reflects the function of the Globals Menu and the Save Field Menus, which are used to define characteristics that are "global"--apply to the entire

forms file. Being off the main stream means that you cannot go directly to an existing save field from the Main Menu as you can go directly to specific field or form definitions. Therefore, you use this other path when you want to examine your save field definitions or modify them. Specifically, you enter G in the selection box on the Main Menu and press ENTER to go to the Globals Menu (described in Section 7). Then press NEXT to display the first Save Field Menu. Thereafter, you use PREV or NEXT to display the defined Save Field Menus, if any. After displaying the first blank Save Field Menu following any defined save fields, NEXT causes the first Form Menu to be displayed.

```

FORMSPEC v.uu.ff Main Menu                                FORMS FILE: ORDFORM1
[S] Enter Selection
Forms file is 2% full
A--Add a form
S--Add a Save field
T--Terminal/Language Selection Menu
G--Go to GLOBALS Menu, OR Go to form [ ] field [ ]
L--List Forms File, OR List form ... [ ]
D--Delete Save field ..... [ ]
   Form ..... [ ]
C--Copy new form name ..... [ ]
   from form ..... [ ]
   from Forms File (opt) ..... [ ]
X--Compile Forms File
   Optional: Fast Forms File [ ]
R--Relate child form [ ] to parent form [ ]
PREV NEXT REFRESH PREV NEXT MAIN/ EXIT
FORM FORM RESUME

```

Figure 6-2. Main Menu

**ADD SAVE FIELDS**

Since save fields do not belong to any particular form, they can be used to transfer data between one form and another. In the case of the forms file, ORDFORM1, you want to transfer the order number, the net total, and the customer name from the ORDER form to the TOTALS form. For this purpose you need three save fields, one for each value to be transferred.

**Action**

To display this menu, run FORMSPEC and request the forms file ORDFORM1 on the Forms File Menu. As shown in Figure 6-2, when the Main Menu is displayed, enter S in the selection box and press ENTER.

```

FORMSPEC v.uu.ff Save Field Menu                                FORMS FILE: ORDFORM1

Save Field Name [SORDNUM ]          Length [6 ]          Data Type [DIG ]
Initial Value   [100001 ]

```

---

```

PREV FORM   NEXT FORM   [ ]   REFRESH   [ ]   PREV   NEXT   MAIN/RESUME   EXIT

```

**Figure 6-3. Save Field for SORDNUM**

This selection causes a blank Save Field Menu to be displayed (not shown).

**Action**

Enter the name (SORDNUM), length (6), and data type (DIG) , and initial value of 100001 to define the save field, as shown in Figure 6-3. Press ENTER.

You should always specify an initial value for fields of type DIG, NUM[n], or IMPn. Character fields default to spaces. You have now defined a save field for calculating and transferring order numbers; it is called SORDNUM and has an initial value of 100001 for the first order number. When you press ENTER to record the first save field, a new blank Save Field Menu is displayed.

**Action**

Using the same method, define the next two save fields, SNAME and STOTNET. Refer to Table 6-1 for a list of the characteristics for the three save fields.

**Table 6-1. Save Field Definitions for ORDFORM1**

Save Field Name	Length	Data Type	Initial Value
SORDNUM	6	DIG	100001
SNAME	42	CHAR	--
STOTNET	10	NUM2	0.00

```

FORMSPEC v.uu.ff Main Menu                                FORMS FILE: ORDFORM1
[G] Enter Selection
Forms file is 1% full
A--Add a form
S--Add a Save field
T--Terminal/Language Selection Menu
G--Go to GLOBALS Menu, OR Go to form [ORDER ] field [ORDNUM ]
L--List Forms File, OR List form ... [ ]
D--Delete Save field ..... [ ]
   Form ..... [ ]
C--Copy new form name ..... [ ]
   from form ..... [ ]
   from Forms File (opt) ..... [ ]
X--Compile Forms File
   Optional: Fast Forms File [ ]
R--Relate child form [ ] to parent form [ ]
PREV  NEXT  REFRESH  PREV  NEXT  MAIN/  EXIT
FORM  FORM  FORM      FORM  FORM  RESUME

```

Figure 6-4. Main Menu

**USE THE SAVE FIELDS**

Now that you have defined the save fields, you need to modify the field definition for the fields that transfer data to, or receive data from, the newly defined save fields. In this exercise, you use the save fields in conjunction with three fields on the ORDER form and three on the TOTALS form. Let's consider the order number first. Not only do you specify that the order number on the ORDER form is to be transferred to the TOTALS form, but you specify that it is to be incremented automatically by 1 until it reaches 999999, after which the order number is reset to 100001.

The SORDNUM save field was given the initial value 100001. This is the first sequential order number. Because the save field contains an initial value, the initial value previously specified for the ORDNUM field is not needed. Instead, you use the SET TO statement to transfer the value in the SORDNUM save field to the ORDNUM field.

**Action**

To modify the ORDNUM field, request the Main Menu and enter the specification shown in Figure Figure 6-4. When you press ENTER, the Field Menu for ORDNUM is displayed.



**Figure 6-5. The Field Menu for ORDNUM in the ORDER Form**

**Action**

On this menu, first change the field type (FType) to D for display-only and the data type (DType) to DIG as shown in Figure 6-5.

The DIG data type allows calculations to be performed on this field. (This is done for consistency--the order number calculations are performed on the SORDNUM save field rather than on the ORDNUM field in either the ORDER form or the TOTALS form.)

**Action**

Delete the initial value since it is no longer needed. Now, press FIELD TOGGLE to go to the Processing Specifications area of the menu and enter

the statement shown in Figure 6-5. Press ENTER to record this field definition.

```

FORMSPEC v.uu.ff Field Menu                                FORM: TOTALS
*****ABC MANUFACTURING*****

Order Number: [ordnum]                                     Date: [ordate      ]
      ^
Num [1  ] Len [6  ] Name [ORDNUM      ] Enh [HI  ] FType [D]  DType [DIG ]
Initial Value [
-----
*** Processing Specifications ***
SET ORDNUM TO SORDNUM

IF SORDNUM GT 999999 THEN SET SORDNUM TO 100001
ELSE SET SORDNUM TO SORDNUM + 1

PREV FORM  NEXT FORM  FIELD TOGGLE  REFRESH          PREV  NEXT  MAIN/RESUME  EXIT

```

Figure 6-6. Field Menu for ORDNUM in the TOTALS Form

### Action

Now, go to the Field Menu for ORDNUM in the TOTALS form. As you did with the similar field in the ORDER form, change the field type to D, the data type to DIG, and delete the initial value. Then, go to the Processing Specifications area of the menu and add the statements shown in Figure 6-6. Press ENTER.

The first time the ORDER form is displayed by ENTRY, the value of SORDNUM is its initial value 100001. Since no change is made to the SORDNUM save field between execution of the first form (ORDER) and the second form (TOTALS), both forms for a single order have the same order number, initially 100001.

The next time the ORDER form is displayed, it is for a new order. As a result of the calculations in the ORDNUM field of the form TOTALS, this number is 1 greater than the previous order number; that is it is 100002. This continues until the order number is incremented to its maximum value of 999999.

Note that the value of SORDNUM is tested before it is incremented. This is done because the two ORDNUM fields and the SORDNUM save field are each six digits long. A value greater than 999999 causes an overflow error because it won't fit in these fields. To prevent such an error, the

order number is reset to its initial value when it reaches the maximum amount the field can hold.

In order to insure that the order number is incremented only in Collect mode, you need a way to prevent it from being incremented in Browse mode whenever ENTER is pressed to correct data in TOTALS. Also, it would be nice if the order numbers could be displayed when each form is initially displayed on the screen. So far, you have had to go into Browse mode to see the current order number. Fortunately, both these problems can be solved by using phases.

**TOPIC: PHASES**

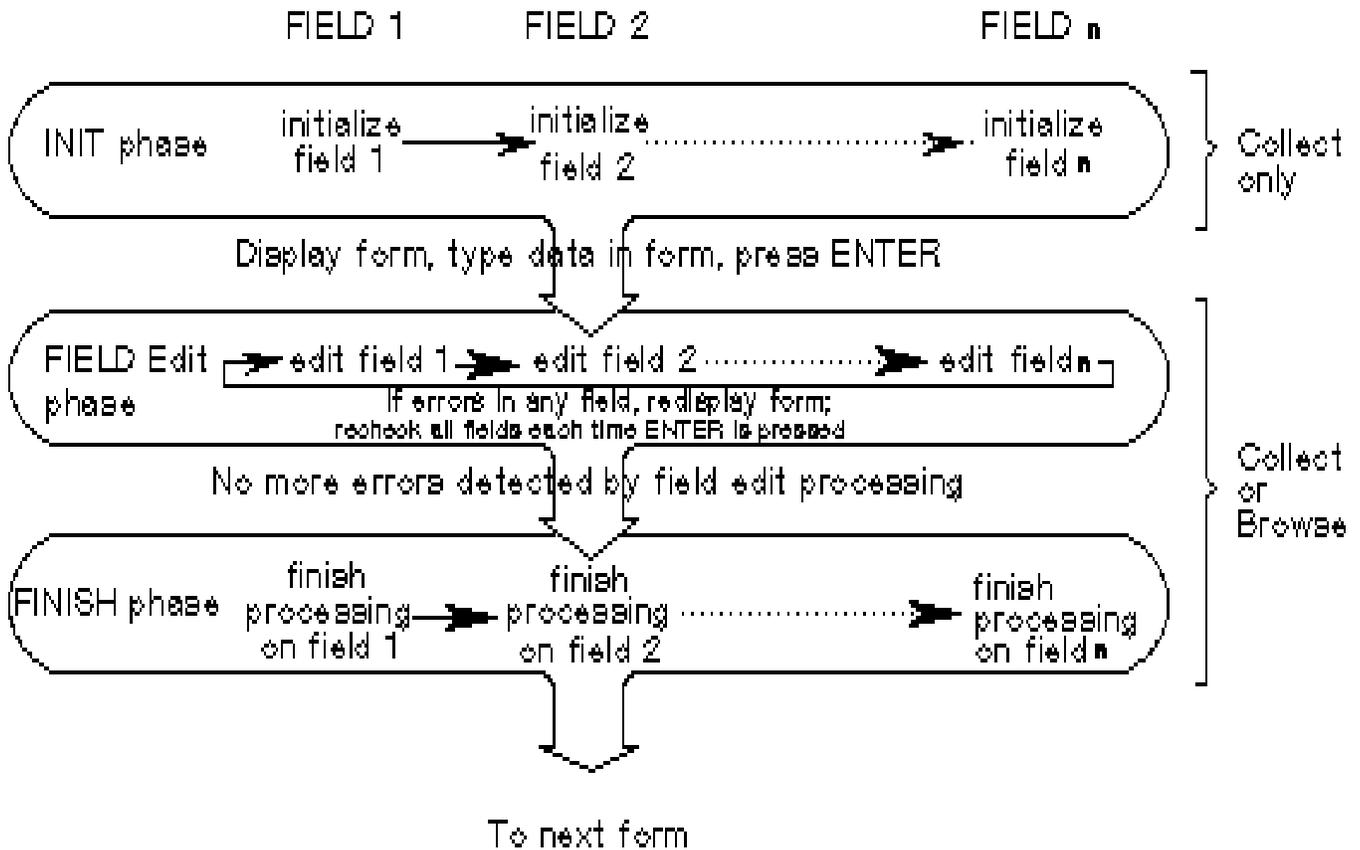
Statements entered in the Processing Specifications area of the Field Menu can be executed in any of three phases: an initialization phase, a field edit phase, and a final phase. Normally, all statements are executed in the field edit phase. But, by the use of phase headers, statements can be executed in either of the other two phases. The three phase headers corresponding to the three possible phases of execution are: INIT, FIELD, and FINISH. If any of these headers precedes a statement or set of statements, those statements are executed in the specified phase.

Basically, the phases determine when a statement is executed. The INIT statements are executed *before* the form is displayed. The FIELD statements are executed when ENTER is pressed and, unlike other statements, may be executed again and again until all data is correct. The FINISH statements are performed *after* all data has been accepted. The rules governing phases are summarized in Table 6-2.

**Table 6-2. Summary of Phases**

Phase	Rules Governing the Phase
INIT	<p>All statements in this phase are performed before the form is displayed.</p> <p>No statements in this phase are executed in Browse mode.</p> <p>Any statements preceding a FIELD header are assumed to be in the initialization phase even if the INIT header is omitted.</p>
FIELD	<p>Statements in this phase are performed after the operator types in data and presses the ENTER key.</p> <p>Statements are executed a field at a time, in screen order, until all fields have been processed.</p> <p>If corrections are entered for any field that has an error, all field edit phase statements are re-executed in screen order.</p> <p>Statements in this phase are executed in Browse mode, and all statements are re-executed when any change to a field is entered.</p> <p>Edit statements (see Section 4) must be executed in field edit</p>

	<p>phase.</p> <p>If there is no phase header, the field edit phase is assumed. The FIELD header must be specified only if there are initialization statements.</p>
FINISH	<p>Statements in this phase are executed only when data entered in all fields in the form have passed the field edits.</p> <p>Statements in this phase are executed in Browse mode after field edit phase statements have been successfully executed.</p> <p>The FINISH header must always precede any statements in this phase.</p>



**Figure 6-7. Diagram of Phase Processing**

Figure 6-7 illustrates graphically how processing is performed on a single form in each of the possible phases. Remember that the initialization phase processing is performed only if any field has initial values, has processing statements preceding a FIELD header, or has processing statements that are preceded by the heading INIT. Statements to be processed in the final phase must be preceded by the header FINISH. If there are no initial values or phase headings, all processing is performed in the field edit phase whether or not the FIELD header is specified.

## Action

For a full discussion of this topic, read "Phases" in Section 4 of the *VPLUS/V Reference Manual*.

**Table 6-3. Specifications to Initialize Order Number and Date**

Form	Field	Processing Specifications
ORDER	ORDNUM	INIT SET ORDNUM TO SORD
	ORDATE	INIT SET ORDATE TO \$TODAY
TOTALS	ORDNUM	INIT SET ORDNUM TO SORDNUM IF SORDNUM GT 999999 THEN SET SORDNUM TO 100001 ELSE SET SORDNUM TO SORDNUM + 1
	ORDATE	INIT SET ORDATE TO \$TODAY

## USE PHASES

Once you have completed the actions described below, the specifications for the ORDNUM and ORDATE fields in each of the two forms should appear as shown in Table 6-3.

## Action

In order to cause the order number on both the ORDER form and the TOTALS form to appear when the forms are initially displayed, you return to the ORDNUM fields for each of these forms and precede the processing statements by the header, INIT. Press ENTER to record the changes.

As a result, the order number is calculated in the save field and then moved to the ORDNUM field before either form is displayed. It is also useful to display the date when a form is displayed.

## Action

This can be accomplished in the ORDATE fields of each form by entering

the phase header INIT before the statement SET TO \$TODAY. Press ENTER to record the change for each field.

Later in this section, you use the INIT phase once again to initialize the total net price for the TOTNET field in TOTALS.

Note that the value of SORDNUM is tested and incremented in the initialization phase, rather than in the final phase, to prevent its being incremented when corrections are entered in Browse mode. It is important to remember that the initialization phase is the only phase not executed in Browse mode.

```
FORMSPEC v.uu.ff Field Menu FORM: ORDER
SHIP TO:
  Name [name          ]
  ^
Num [2 ] Len [42 ] Name [NAME          ] Enh [HI ] FType [R] DType [CHAR]
Initial Value [
*** Processing Specifications ***
SET SNAME TO NAME

PREV FORM  NEXT FORM  FIELD TOGGLE  REFRESH
PREV       NEXT       MAIN/RESUME  EXIT
```

Figure 6-8. Field Menu for NAME in the ORDER Form

### TRANSFER VALUES BETWEEN FORMS

So far you transferred the same value from a save field to fields in two different forms. Now, you use save fields to transfer values entered on one form to another form. The two save fields, SNAME and STOTNET are used for this purpose. The value of NAME in the ORDER form is transferred to the SNAME save field. Then the value in SNAME is transferred to the NAME field in TOTALS. Similarly, the value in TPRICE of the ORDER form is transferred to the STOTNET save field, and the value in STOTNET is subsequently moved to the TOTNET field of TOTALS. To perform these transfers, SET statements must be added to the name and net total fields of both the ORDER form and the TOTALS form. Start with the NAME field in the ORDER form.

**Action**

Find the Field Menu for NAME and change the field type to R for required. Then press FIELD TOGGLE to enter the processing specifications shown in Figure 6-8. Press ENTER to record the changes.

Making NAME a required field insures that a customer name is actually entered. Though not necessary for the transfer to TOTALS, it is good practice to associate a customer name with each order. The SET statement is executed in the field edit phase since no heading is specified. This means that if the name is changed, the change is also recorded in the save field.



**Figure 6-9. The Field Menu for NAME in the TOTALS Form**

Now, transfer this name to the customer name field in TOTALS.

**Action**

Find the Field Menu for NAME in form TOTALS, then enter the specifications shown in Figure 6-9. Note that the field type is changed to D for display-only so the operator cannot change the name on this form. Press ENTER to record your changes.

```

FORMSPEC v.uu.ff Field Menu                                FORM: ORDER
QTY  PART NO.      DESCRIPTION                                UNIT  TOTAL
                                           PRICE  PRICE
qty.. partnum..  description.....  uprice. tprice.
                                           ^
Num [15 ] Len [7  ] Name [TPRICE ] Enh [HI ] FType [D] DType [NUM2]
Initial Value [

```

\*\*\* Processing Specifications \*\*\*

```

SET TPRICE TO QTY * UPRICE
SET STOTNET TO TPRICE

```

PREV FORM	NEXT FORM	FIELD TOGGLE	REFRESH	PREV	NEXT	MAIN/ RESUME	EXIT
--------------	--------------	-----------------	---------	------	------	-----------------	------

**Figure 6-10. Field Menu for TPRICE**

Now, make similar changes to the TPRICE field of the ORDER form and the TOTNET field of the TOTALS form.

### Action

Add a statement to store the value of TPRICE in the STOTNET save field, as shown in Figure 6-10.

Note that the order in which the processing specification statements are entered is important. The statement to move the value to the save field must follow the statement that calculates that value. Since both statements are executed in the field edit phase, any change to the values of either QTY or UPRICE in Browse mode change the value of TPRICE, which is then moved to the STOTNET save field.

Total Net Price: [totnet ]

```

Num [3 ] Len [10 ] Name [TOTNET ] Enh [HI ] FType [D] DType [NUM2]
Initial Value [ ]

```

\*\*\* Processing Specifications \*\*\*

INIT

SET TOTNET TO STOTNET

FIELD

SET TOTNET TO STOTNET

PREV  
FORMNEXT  
FORMFIELD  
TOGGLE

REFRESH

PREV

NEXT

MAIN/  
RESUME

EXIT

**Figure 6-11. The Field Menu for TOTNET in the TOTALS Form**

The last change is to the TOTNET field in form TOTALS.

**Action**

First, change the field type to D for display-only and then add the statements (shown in Figure 6-11) to transfer the calculated net price from the ORDER form. You should delete the existing formatting statements (JUSTIFY RIGHT and STRIP LEADING "0") since the SET statement causes this formatting to be performed automatically. Press ENTER to record the changes.

The SET statement must also appear in the FIELD phase to insure that the net total is updated if changes are made in Browse mode. Statements in the INIT phase are only executed once, therefore, if net total was changed on the ORDERS form the new total would be carried over to the TOTALS form by the FIELD statement.

When the TOTALS form is displayed, the total net price calculated on the previous form (ORDER) is shown in the TOTNET field. Then, when the user presses ENTER, the remaining values in the form are calculated, and all values are written to the batch file. The calculated values do not appear during data collection, but only when the batch file is browsed.

[X] Enter Selection

```

A--Add a form
S--Add a Save field
T--Terminal/Language Selection Menu
G--Go to GLOBALS Menu, OR Go to form [ ] field [ ]

L--List Forms File, OR List form ... [ ]

D--Delete Save field ..... [ ]
   Form ..... [ ]

C--Copy new form name ..... [ ]
   from form ..... [ ]
   from Forms File (opt) ..... [ ]

X--Compile Forms File
   Optional: Fast Forms File [ ]

R--Relate child form [ ] to parent form [ ]
    
```

PREV FORM    NEXT FORM    REFRESH    PREV    NEXT    MAIN/RESUME    EXIT

Figure 6-12. Main Menu, Compile Selection

**Action**

Compile the forms file to record the new specifications in machine-readable form, as shown in Figure 6-12.

Table 6-4 and Table 6-5 summarize all the current field specifications for the ORDER and TOTALS forms. (Refer to Table 6-1 for a summary of the save field specifications.)

Table 6-4. Summary of Current Fields in ORDER

Num	Name	FType	DType	Processing Specifications	Initial Value
8	ORDNUM	D	DIG	INIT SET ORDNUM TO SORDNUM	--
1	ORDATE	D	MDY	INIT SET ORDATE TO \$TODAY	--
2	NAME	R	CHAR	SET SNAME TO NAME	--
3	ADDRESS	O	CHAR	--	--
4	CITY	O	CHAR	--	--

5	ST	O	CHAR	IN \$STATE	--
6	ZIP	O	CHAR	MATCH dddd[-dddd]	--
7	PHONE	O	CHAR	MATCH [(ddd)]b*ddd-dddd	--
9	QTY	R	DIG	GE 1	--
10	PARTNUM	R	CHAR	MATCH Pudd-???	--
11	DESCRIPTION	O	CHAR	--	--
12	UPRICE	R	NUM2	JUSTIFY RIGHT STRIP LEADING "0"	--
13	TPRICE	D	NUM2	SET TPRICE TO QTY * UPRICE SET STOTNET TO TPRICE	--

**Table 6-5. Summary of Current Fields in TOTALS**

Field					
Num	Name	FType	DType	Processing Specifications	Value
1	ORDNUM	D	DIG	INIT SET ORDNUM TO SORDNUM IF SORDNUM GT 999999 THEN SET SORDNUM TO 100001 ELSE SET SORDNUM TO SORDNUM + 1	--
2	ORDATE	D	MDY	INIT SET ORDATE TO \$TODAY	--
3	TOTNET	D	NUM2	INIT SET TOTNET TO STOTNET FIELD SET TOTNET TO STOTNET	--
4	DISCNT_PRCNT	O	NUM0	--	10
5	DISCNT	D	NUM2	IF TOTNET GE 1000 THEN SET DISCNT TO DISCNT_PRCNT % TOTNET ELSE SET DISCNT TO 0	--
6	SUB	D	NUM2	SET SUB TO TOTNET - DISCNT	--
7	TAX_RATE	R	NUM2	--	6.25
8	TAX	D	NUM2	SET TAX TO TAX_RATE % SUB	--
9	SHIP	O	NUM2	JUSTIFY RIGHT STRIP LEADING "0"	.00
10	TOTAL	D	NUM2	SET TOTAL TO SUB + TAX + SHIP	--
11	NAME	D	CHAR	SET NAME TO SNAME	--

:RUN ENTRY.PUB.SYS

HP32209v.uu.ff ENTRY (C) HEWLETT-PACKARD CO. 1986

ENTER FORMS FILE NAME AND PRESS RETURN: ORDFORM1

ENTER BATCH FILE NAME AND PRESS RETURN: DATA6

\*\*\*\*\*ABC MANUFACTURING\*\*\*\*\*

Order Number: [100001]

Date: [09/02/86]

Name [RANDOLPH BROTHERS]  
Address [14327 University Place]  
City [Santa Clara] State [Ca]  
Zip [95050] Telephone [(408)327-6600]

QTY	PART NO.	DESCRIPTION	UNIT PRICE	TOTAL PRICE
5	PA123-abc		10.50	

ENTRY v.uu.ff Batch Record #1 Mode: Collect  
HEAD PRINT REFRESH NEXT BROWSE EXIT  
FORM

Figure 6-13. Run ENTRY and the Completed ORDER Form, Record No. 1

#### REVIEW VALUES TRANSFERRED BETWEEN FORMS

Now use ENTRY to see the order number incremented and other values transferred between forms.

#### Action

Run ENTRY and specify the forms file, ORDFORM1. Use a new name for the batch file since ORDFORM1 has been changed and recompiled.

ENTRY will display the first form, ORDER. Note that the order number is shown as 100001 and the date as today's date in the form mm/dd/yy. The cursor will be positioned at the first field in which you can enter any data, in this case, the customer name field. Since this field is now required, you must enter a name.

**Action**

Type in the name and the remaining values for the order, as shown in Figure 6-13, then press ENTER.

```
-
          *****ABC MANUFACTURING*****
Order Number: [100001]                      Date: [09/02/86 ]
                                           Total Net Price: [ 52.50]
Discount % [10]                            Discount: [      ]
                                           Subtotal: [      ]
Tax Rate [6.25]                             Tax: [      ]
                                           Shipping: [.00 ]
                                           TOTAL: [      ]
Customer Name [      ]
ENTRY v.uu.ff      Batch Record #2      Mode: Collect
HEAD | PRINT | REFRESH | NEXT | BROWSE | EXIT
FORM |      |      | FORM |      |      |
```

**Figure 6-14. TOTALS Form, Record No. 2**

When you press ENTER to record the typed values, the system calculates the total price. It then displays the next form, TOTALS, as shown in Figure 6-14.

The order number is the same as the number displayed on the previous form, ORDER, since this form contains the totals for the ORDER form. The order number is displayed since it was transferred from the save field during the initialization phase (INIT). The date too is displayed because it was calculated in the initialization phase.

Of the two remaining values transferred from the ORDER form, only the total net price is shown when the form is displayed. This is because it, like the order number, is moved from the save field in the initialization phase. The customer name, on the other hand, is set in the field edit phase and is not moved until the user presses ENTER.

The first field in which you can enter data is the Discount % field. You may change the discount percent from 10 to some other value, and you may change the tax rate and the shipping cost. These are the only three values that may be entered on this form; the other values are all display-only. If you do not want to change these three values, simply

press ENTER to record all the totals.

### Action

For this exercise, make no changes and press ENTER to record the totals in the batch file and then display the next form.

```

-
*****ABC MANUFACTURING*****
Order Number: [100001] Date: [09/02/86 ]

SHIP TO:
Name [RANDOLPH BROTHERS ]
Address [14327 University Place ]
City [Santa Clara ] State [Ca]
Zip [95050 ] Telephone [(408)327-6600 ]

QTY PART NO. DESCRIPTION UNIT TOTAL
PRICE PRICE
5 PA123-abc 10.50 52.50

ENTRY v.uu.ff Batch Record #1 Mode: Browse
FIRST DELETED PRINT REFRESH PREV NEXT COLLECT EXIT
REC REC REC REC REC REC REC REC
```

Figure 6-15. Browse Mode, Record No. 1

### REVIEW AUTOMATIC VALUES

Before entering another order, let's see what values have been calculated by the system.

### Action

To do this, you must press BROWSE to go into Browse mode and then press FIRST REC to display the first record of the current order, as shown in Figure 6-15.

Note that the total price is now displayed, and that both the unit price and total price are right justified in the field.

```

-
*****ABC MANUFACTURING*****
Order Number: [100001] Date: [09/02/86 ]

Total Net Price: [ 52.50]
Discount % [10] Discount: [ .00]
Subtotal: [ 52.50]
Tax Rate [6.25] Tax: [ 3.28]
Shipping: [ .00]
TOTAL: [ 55.78]

Customer Name [RANDOLPH BROTHERS ]

ENTRY v.uu.ff Batch Record #2 Mode: Browse
FIRST DELETE PRINT REFRESH PREV NEXT COLLECT EXIT
REC REC REC REC REC REC REC REC

```

Figure 6-16. Browse Mode, Record No. 2

**Action**

In order to see the data on the next form, press NEXT REC.

The values calculated on the TOTALS form are displayed, as shown in Figure 6-16.

\*\*\*\*\*ABC MANUFACTURING\*\*\*\*\*

Order Number: [100002]

Date: [09/02/86]

SHIP TO:

Name [BOND FABRICS, INC.]  
Address [5570 15th Street]  
City [San Jose] State [Ca]  
Zip [95162] Telephone [(408)334-7790]

QTY	PART NO.	DESCRIPTION	UNIT PRICE	TOTAL PRICE
100	PA123-xh1		10.50	

ENTRY v.uu.ff Batch Record #3 Mode: Collect  
HEAD PRINT REFRESH NEXT BROWSE EXIT  
FORM FORM

Figure 6-17. Browse Mode, Record No. 3

### Action

After browsing the first order (order number 100001), return to Collect mode by pressing COLLECT.

A new form is displayed on which you can enter the next order. This form shows the order number as 100002 since the order number was incremented automatically when the TOTALS form for the previous order was initialized.

### Action

Type data on this form as shown in Figure 6-17, then press ENTER.

As before, the total price (1050) is calculated by the system.

:RUN ENTRY.PUB.SYS

HP32209v.uu.ff ENTRY (C) HEWLETT-PACKARD CO. 1986

ENTER FORMS FILE NAME AND PRESS RETURN: ORDFORM1

ENTER BATCH FILE NAME AND PRESS RETURN: DATA6

```

-
          *****ABC MANUFACTURING*****
Order Number: [100002]                      Date: [09/02/86 ]

Total Net Price: [ 1050.00]
Discount % [10]                               Discount: [ ]
Subtotal: [ ]
Tax Rate [6.25]                               Tax: [ ]
Shipping: [.00]
TOTAL: [ ]

Customer Name [ ]

ENTRY v.uu.ff      Batch Record #4      Mode: Collect
HEAD              PRINT  REFRESH              NEXT  BROWSE  EXIT
FORM

```

Figure 6-18. TOTALS Form, Record No. 4

#### TERMINATE AND RESUME ENTRY IN MID-ORDER

##### Action

After entering the data on the first form of the order number 100002, press EXIT to terminate ENTRY.

You return to MPE control, as signaled by the colon prompt (:). Now, run ENTRY again with the same files you specified at the start of this exercise.

##### Action

Run ENTRY. Then, when ENTRY asks for the forms file and batch file names, enter the files you have been using; ORDFORM1 and DATA6.

ENTRY resumes operation where you left off exactly as if you had not terminated the program. That is, it issues the next sequential form in Collect mode. Note that the order number and the total net price have both been saved through the program termination and now are displayed on the TOTALS form, as shown in Figure 6-18.

##### Action

Press ENTER to calculate the totals for the order entered in the previous session.

```

-
*****ABC MANUFACTURING*****
Order Number: [100002]           Date: [09/02/86 ]

Total Net Price: [ 1050.00]
Discount % [10]           Discount: [ 105.00]
Subtotal: [ 945.00]
Tax Rate [6.25]           Tax: [ 59.06]
Shipping: [ .00]
TOTAL: [ 1004.06]

Customer Name [BOND FABRICS, INC. ]

ENTRY v.uu.ff           Batch Record #4           Mode: Browse
FIRST  DELETE  PRINT  REFRESH  PREV  NEXT  COLLECT  EXIT
REC    REC

```

Figure 6-19. Browse Mode, Record No. 4

### Action

Press BROWSE.

The TOTAL form you just entered is displayed with all the values calculated. The total net price and the customer name were transferred from the form on which they were entered before you pressed EXIT to terminate.

Similarly, if you terminate between orders, the order number is incremented and displayed on the next form when you resume execution with the same batch file.

```

-
*****ABC MANUFACTURING*****
Order Number: [100001] Date: [09/02/86 ]

SHIP TO:
Name [RUDOLPH BROTHERS ]
Address [14327 University Place ]
City [Santa Clara ] State [Ca]
Zip [95050 ] Telephone [(408)327-6600 ]

QTY PART NO. DESCRIPTION UNIT TOTAL
PRICE PRICE
5 PA123-abc 10.50 52.50

ENTRY v.uu.ff Batch Record #1 Mode: Browse
FIRST DELETE PRINT REFRESH PREV NEXT COLLECT EXIT
REC REC REC REC REC REC REC REC

```

Figure 6-20. Browse Mode, Corrected Record No. 1

#### MODIFY DATA TRANSFERRED BETWEEN RECORDS

To modify any data that has been written to a batch file, you must be in Browse mode. Then you locate the record and correct it. When you press ENTER, the corrected data is written over the previous data in the record. If this data appears only in one record or does not affect any other record, that is all you need to do. For example, if you change a customer address entered on the ORDER form, you need not worry about the totals transferred to the TOTALS form.

If, however, you change a value that is transferred to another record, you must locate that record and press ENTER to record the new value. For example, if you change the customer name in Record #1 of the current batch file, you must make sure the name is also changed on Record #2.

In this exercise, suppose you want to change a customer name from RANDOLPH BROTHERS to RUDOLPH BROTHERS.

#### Action

First, locate the record on which the name was originally entered, in this case, Record #1. Then, change the name and press ENTER, as shown in Figure 6-20.

Since the customer name is always entered on this form and then transferred via a save field to the totals form, you must also change this value on the next record. But you cannot change the name on the

TOTALS form directly--it is a display-only field protected from data entry.

\*\*\*\*\*ABC MANUFACTURING\*\*\*\*\*

Order Number: [100001] Date: [09/02/86]

	Total Net Price:	[ 52.50]
Discount % [10]	Discount:	[ .00]
	Subtotal:	[ 52.50]
Tax Rate [6.25]	Tax:	[ 3.28]
	Shipping:	[ .00]
	TOTAL:	[ 55.78]

Customer Name [RANDOLPH BROTHERS]

ENTRY v.uu.ff	Batch Record #2			Mode: Browse			
FIRST REC	DELETE REC	PRINT	REFRESH	PREV REC	NEXT REC	COLLECT	EXIT

Figure 6-21. Browse Mode, Record No. 2

When you press ENTER to record the correct name on the ORDER form, the next record is displayed. This record still contains the old incorrect customer name, as shown in Figure 6-21.

Although you cannot change the customer name directly on this form, you can force the new name to be transferred by pressing ENTER when the TOTALS form is displayed. Pressing ENTER causes the SET NAME TO SNAME statement in the NAME field to be executed, thereby transferring the new name from the SNAME save field.

```

-
*****ABC MANUFACTURING*****
Order Number: [100001] Date: [09/02/86 ]

Total Net Price: [ 52.50]
Discount % [10] Discount: [ .00]
Subtotal: [ 52.50]
Tax Rate [6.25] Tax: [ 3.28]
Shipping: [ .00]
TOTAL: [ 55.78]

Customer Name [RUDOLPH BROTHERS ]

ENTRY v.uu.ff Batch Record #2 Mode: Browse
FIRST DELETE PRINT REFRESH PREV NEXT COLLECT EXIT
REC REC REC REC REC REC REC REC

```

Figure 6-22. Browse Mode, Corrected Record No. 2

### Action

So, press ENTER and then press PREV REC to return to Record #2. You see that the name has been corrected, as shown in Figure 6-22.

Similarly, if you change the quantity or unit price of an order recorded on the ORDER form, you must then press ENTER on the corresponding TOTALS record to make sure the order total is correct.

### Action

The exercise is complete; press EXIT.

## Chapter 7 Using Save Fields and Frozen/Repeating Forms

The two forms defined so far in the forms file ORDFORM1 allow only a one line item per order. In order to allow a number of items to be entered on a single order, you separate the first form (ORDER) into two forms: the first, containing header information, is frozen on the screen; the second, containing one line of order information, is repeated and appended to the frozen form. This allows a variable number of parts to be entered on a single order.

This section also shows how to accumulate the totals for an order into a save field and then transfer the final total to the TOTALS form. This section illustrates the following features:

Using FORMSPEC:

- \* Copy and Delete forms (Main Menu)
- \* Freeze form, append next form (Form
- \* Change form layout of copied form.
- \* Repeat, append current form (Form Menu).
- \* Specify head form (Globals Menu).
- \* Change window line (Globals Menu).
- \* Accumulate totals (Field Menu).
- \* Change accumulated values.
- \* Change form sequence programmatically.

Using ENTRY:

- \* Display frozen and repeating, appending form.
- \* Use NEXT FORM to stop repeat.
- \* Automatic termination of repeat.
- \* Modify accumulated values.
- \* Use DELEE REC to delete record.

```

-
                                     *****ABC MANUFACTURING*****

Order Number: [ordnum]                               Date: [ordate] ]

SHIP TO:
  Name [name ]
Address [address ]
  City [city ] State [st]
  Zip [zip ] Telephone [phone ]

QTY   PART NO.   DESCRIPTION                               UNIT   TOTAL
qtym.. partnum.. description.....                PRICE  PRICE

PREV  NEXT  REFRESH  PREV  NEXT  MAIN/  EXIT
FORM  FORM  FORM      FORM  FORM  RESUME

```

Figure 7-1. The Form Layout for the ORDER Form

#### CHANGE FORMS FILE

In the ORDFORM1 forms file, the ORDER form, as shown in Figure 7-1, contains customer information plus one line on which an order can be entered. This restricts each order to one part number--a more practical design is to have *multiple* order lines as well as the customer information. In this exercise you separate the ORDER form into two forms, one called ORDER\_HEAD, that contains all the customer information from the ORDER form, and another form called ORDER\_LINE, that contains only the line with order information. You define the ORDER\_HEAD form as frozen and the ORDER\_LINE form as a repeating form appended to itself. The result, when you run ENTRY, is that the customer information is frozen on the screen with order lines appended to it each time the user adds an order.

Since the ORDER\_LINE form is repeated, new data can be entered until there are no more parts to order. At that point, the user terminates the repeat by pressing NEXT FORM in ENTRY. Although the next form is identical to the existing TOTALS form, in this exercise let's give it a new name (ORDER\_TOTALS) to be consistent with the form names for the two versions of the ORDER form. The result of redefining the forms is a file with three forms:

ORDER\_HEAD            Contains the order number, date, customer name, and address, and headings for the order line (same as the ORDER form except the order line is omitted).

ORDER\_LINE            Contains fields in which to enter quantity, part number, description, unit price, and total price of a single part (order line from the ORDER form).

ORDER\_TOTALS         Contains totals for entire order (identical to the TOTALS form).

Using FORMSPEC, you can copy forms within an existing file or to a new forms file. Therefore, rather than typing in the two forms, ORDER\_HEAD and ORDER\_LINE, let's copy the ORDER form twice within the ORDFORM1 forms file. You can then modify the two copies of the ORDER form and give the TOTALS form a new name, ORDER\_TOTALS.

Since forms are kept in alphabetical order in the forms file, the ORDER\_HEAD form will be the first form in the file. By default, the first form is treated as the head form and is the form displayed first by ENTRY.



Figure 7-2. The Main Menu

**COPY FORMS**

Assuming you have run FORMSPEC and opened a forms file, the steps to copy a form are:

1. On the Main Menu, specify a C in the selection box, the name of the form to be copied, and a name for the new form. Optionally, you may need to specify the name of the forms file containing the





## **FREEZE FORM, APPEND NEXT FORM**

The ORDER\_HEAD form contains the date, the customer identification, the order number, and the item headings for the next form, ORDER\_LINE. Because this form contains the headings for the ORDER\_LINE form, it should be "frozen" on the screen while the order line is appended to it. A frozen form stays on the screen when the next form is displayed. It also stays on the screen if the next form is repeated so many times that there is no more room on the screen. In this case, the first repeating form is rolled off the screen under the frozen form, but the frozen form remains.

### **Action**

Use the Main Menu (option G) to go to the Form Menu for the ORDER\_HEAD form.

When the Form Menu for ORDER\_HEAD is displayed, the repeat option is, by default, set to N for no repeat.

### **Action**

Leave the repeat option as is, but change the next form option to F for freeze and append next form. Change the next form name from the TOTALS to the form name ORDER\_LINE and change the comment to match the one shown in Figure 7-4; press ENTER.

Note that the comment indicates that this form is the first form executed. This is actually the case because the forms in a forms file are in alphabetical order and ORDER\_HEAD is first of the forms currently defined. However, the comment would not be true if the forms file was changed to contain a form with a name that came before ORDER\_HEAD. Later, you will insure that this form is always the first form executed, regardless of the sequence of forms in the forms file, by using the Globals Menu to specify ORDER\_HEAD as the "head" form in the file.

\*\*\*\*\*ABC MANUFACTURING\*\*\*\*\*

Order Number: [ordnum] Date: [ordate ]

SHIP TO:  
Name [name ]  
Address [address ]  
City [city ] State [st]  
Zip [zip ] Telephone [phone ]

QTY	PART NO.	DESCRIPTION	UNIT PRICE	TOTAL PRICE
qty..	partnum..	description.....	uprice.	tprice.



Figure 7-5. The Form Layout Menu for ORDER\_HEAD

**CHANGE FORM LAYOUT FOR HEAD FORM**

When you press ENTER, the form layout for ORDER\_HEAD is displayed. It is still an identical copy of the deleted form, ORDER, and appears as shown in Figure 7-5. The only change you make to the form layout is to delete the line on which order information is entered, the last line on the form.

**Action**

Use the terminal edit keys to delete the line in the ORDER\_HEAD form layout on which order information is entered. Press ENTER to record the modified form layout.

```

FORMSPEC v.uu.ff Form Menu                                FORMS FILE: ORDFORM1

Form Name [ORDER_LINE ]
Repeat Option [A]
             N--No Repeat
             A--Repeat, appending
             R--Repeat, overlaying

Next Form [C]                                           Name [ORDER_TOTALS ]
             C--Clear before Next Form
             A--Append Next Form
             F--Freeze, then append Next Form

Function Key Label [ ]
             Y--Define form level function key labels.

Reproduced from [ ] (opt)

Comments [2nd form-repeated automatically,appended to itself]

PREV  NEXT  REFRESH  PREV  NEXT  MAIN/  EXIT
FORM  FORM  FORM      FORM  FORM  RESUME

```

Figure 7-6. The Form Menu for ORDER\_LINE

**REPEAT CURRENT FORM, APPEND IT TO ITSELF**

Before changing any field characteristics on this form, you complete the form definition.

**Action**

Press NEXT FORM to go directly to the Form Menu for ORDER\_LINE. Change the Repeat Option to A for repeat, appending. Leave the value C in the Next Form option, and change the next form name from TOTALS to ORDER\_TOTALS. The Form Menu for the ORDER\_LINE form should look like Figure 7-6; press ENTER.

Remember that the form currently named TOTALS is to be renamed ORDER\_TOTALS for consistency with the other form names.

```

-
                *****ABC MANUFACTURING*****

Order Number: [ordnum]                               Date: [ordate      ]

SHIP TO:
  Name [name                ]
Address [address             ]
  City [city                 ] State [st]
  Zip [zip       ] Telephone [phone    ]

QTY   PART NO.   DESCRIPTION                               UNIT   TOTAL
      .          .          .                               PRICE  PRICE
qty.. partnum.. description..... uprice.  tprice.

PREV  NEXT      REFRESH      PREV  NEXT  MAIN/  EXIT
FORM  FORM                                     RESUME

```

Figure 7-7. The Form Layout Menu for ORDER\_LINE

#### DESIGN LAYOUT FOR NEW FORM

The form for ORDER\_LINE appears exactly like the form layout for ORDER, as shown in Figure 7-7, since it was copied from ORDER. Now, you want to delete every line except the line on which orders are entered.

#### Action

Delete all lines but the order information line (the last line). Press ENTER to record the form for ORDER\_LINE.

qty.. partnum.. description..... uprice. tprice.



**Figure 7-8. The Form Layout Menu for ORDER\_LINE, completed**

The form for ORDER\_LINE should look like Figure 7-8. This method of creating new forms from an existing form may appear complicated on paper, but it is far more efficient than designing the new form from scratch. Because the headings for the fields are on one form and the fields themselves on another, it could have been difficult to align the two forms correctly, especially when the fields are delimited by invisible delimiters. By creating one correctly aligned form and then copying it twice, you insure that the headings and the fields line up.

```

FORMSPEC v.uu.ff Form Menu                                FORMS FILE: ORDFORM1

Form Name [ORDER_TOTALS ]
Repeat Option [N]
             N--No Repeat
             A--Repeat, appending
             R--Repeat, overlaying

Next Form [C]                                     Name [$HEAD ]
             C--Clear before Next Form
             A--Append Next Form
             F--Freeze, then append Next Form

Function Key Label [ ]
             Y--Define form level function key labels.

Reproduced from [ ] (opt)

Comments [3rd form of a 3-part order transaction ]

PREV FORM  NEXT FORM  REFRESH  PREV  NEXT  MAIN/RESUME  EXIT

```

Figure 7-9. The Form Menu for ORDER\_TOTALS

**CHANGE FORM NAME**

The final change to the form definitions for the new version of ORDFORM1 is to change the name of TOTALS to ORDER\_TOTALS.

**Action**

Press NEXT FORM to get to the Form Menu for TOTALS. Change the form name to ORDER\_TOTALS and change the comment as shown in Figure 7-9; the other options can remain as specified. Press ENTER to record these changes.

No changes are required to the form layout for this form. Except for the new name, it is identical to the TOTALS form.

With ENTRY, when the user presses NEXT FORM to stop the repeat of the form ORDER\_LINE, the screen is cleared and the ORDER\_TOTALS form is displayed. After totals are entered on ORDER\_TOTALS, ENTRY returns to the head form, as specified by the Next Form Name in Figure 7-9.

Now you use the Globals Menu to specify the head form--the first form displayed when the forms file is executed.

```

FORMSPEC v.uu.ff Field Menu                                FORM: ORDER
QTY  PART NO.      DESCRIPTION                               UNIT  TOTAL
                                           PRICE  PRICE
qty.. partnum..  description.....  uprice.  tprice.
                                           ^
Num [15 ] Len [7  ] Name [TPRICE ] Enh [HI ] FType [D]  DType [NUM2]
Initial Value [

```

\*\*\* Processing Specifications \*\*\*

```

SET TPRICE TO QTY * UPRICE
SET STOTNET TO TPRICE

```

```

PREV FORM  NEXT FORM  FIELD TOGGLE  REFRESH          PREV  NEXT  MAIN/RESUME  EXIT

```

Figure 7-10. The Main Menu, Globals Menu Option

**USE GLOBALS MENU TO IDENTIFY HEAD FORM**

**Action**

Return to Main Menu by pressing MAIN. When the Main Menu is displayed, enter G in the selection box to request the Globals Menu, as shown in Figure 7-10; press ENTER.

Total Net Price: [totnet ]

Num [3 ] Len [10 ] Name [TOTNET ] Enh [HI ] FType [D] DType [NUM2]  
Initial Value [ ]

\*\*\* Processing Specifications \*\*\*

INIT  
SET TOTNET TO STOTNET

FIELD  
SET TOTNET TO STOTNET

PREV FORM    NEXT FORM    FIELD TOGGLE    REFRESH                    PREV    NEXT    MAIN/RESUME    EXIT

**Figure 7-11. The Globals Menu**

The Globals Menu allows you to specify the name of the first form to be displayed. It also allows you to change other global characteristics of the forms file. There are defaults provided for all the global characteristics, which you can accept or change as desired. Specifically, you can change the field, error, or window line enhancements and you can direct the window line to any line on the screen.

**Action**

To indicate the sequence in which ENTRY will display the ORDFORM1 forms, specify ORDER\_HEAD in the Head Form Name field. Also let's change the window line from its current position as the bottom (24th) line of the screen to the top (1st) line by entering 1 in the Window Display Line field. Press ENTER to record these global changes to the forms file.

When ENTRY uses the form ORDFORM1, it automatically displays ORDER\_HEAD first. The error and status line previously shown on the bottom line of any form displayed by ENTRY is now displayed as the top line of each form.

[X] Enter Selection

```

A--Add a form
S--Add a Save field
T--Terminal/Language Selection Menu
G--Go to GLOBALS Menu, OR Go to form [ ] field [ ]

L--List Forms File, OR List form ... [ ]

D--Delete Save field ..... [ ]
   Form ..... [ ]

C--Copy new form name ..... [ ]
   from form ..... [ ]
   from Forms File (opt) ..... [ ]

X--Compile Forms File
   Optional: Fast Forms File [ ]

R--Relate child form [ ] to parent form [ ]

```

PREV  
FORM

NEXT  
FORM

REFRESH

PREV

NEXT

MAIN/  
RESUME

EXIT

Figure 7-12. The Field Menu for TPRICE

### ACCUMULATE TOTALS

In the previous version of ORDFORM1, the total price for the order was calculated and stored in the TPRICE field. This value was then transferred to the TOTNET field in the next form. Now that the order line is a repeating form, you must accumulate the individual values calculated for the TPRICE field in the ORDER\_LINE form and transfer the total amount of all the orders to the TOTNET field in the ORDER\_TOTALS form.

The field specifications on TPRICE are currently specified as:

```

SET TPRICE TO QTY * UPRICE
SET STOTNET

```

These statements calculate a value for the order line and then save this value in the STOTNET save field, which allows you to store the total price calculated for a single order line. The next step is to accumulate the total price for all order lines.

### Action

Change the second statement on the Field Menu for TPRICE as shown in Figure 7-12. Press ENTER.

This statement increments the current value of the STOTNET save field by the calculated value in TPRICE. As the ORDER\_LINE form is repeated a new

value for TPRICE is calculated and this value is added to the total value for the order in STOTNET.

\*\*\*\*\*ABC MANUFACTURING\*\*\*\*\*

Order Number: [100001]

Date: [09/02/86]

Name [RANDOLPH BROTHERS]  
Address [14327 University Place]  
City [Santa Clara] State [Ca]  
Zip [95050] Telephone [(408)327-6500]

QTY	PART NO.	DESCRIPTION	UNIT PRICE	TOTAL PRICE
5	PA123-abc		10.50	

ENTRY v.uu.ff      Batch Record #1      Mode: Collect  
HEAD      PRINT      REFRESH      NEXT      BROWSE      EXIT  
FORM      FORM

Figure 7-13. The Field Menu for ORDNUM

However, in order for the accumulation specified for TPRICE to work, you must reset the value of STOTNET to zero at the beginning of each order. In the first place, STOTNET must have an initial value for it to be used in a calculation; and in the second place, you only want to store the totals for a single order, not keep incrementing the value as the user progresses from one order to another.

**Action**

Return to the Field Menu for the first field in the order, ORDNUM in ORDER\_HEAD, and add a statement to initialize STOTNET to zero, as shown in Figure 7-13. Press ENTER.

The STOTNET save field is now cleared for each new order. Then, as totals are calculated for individual order lines, STOTNET is incremented.

Finally, in the initialization phase of the last form, ORDER\_TOTALS, the value accumulated into STOTNET is moved to TOTNET and displayed.

**TOPIC: CHANGES TO ACCUMULATED VALUES**

As specified so far, if the quantity or unit price of any order line is changed in Browse mode, the changed value is added to the existing accumulated value in STOTNET making the total incorrect. To illustrate,

suppose the following three order lines have been entered on form ORDER\_LINE in Collect mode:

```
1ST QTY = 2  UPRICE = 10.50  TPRICE = 21.00    (STOTNET = 21.00)
2ND QTY = 4  UPRICE = 15.00  TPRICE = 60.00    (STOTNET = 81.00)
3RD QTY = 3  UPRICE = 14.00  TPRICE = 42.00    (STOTNET = 123.00)
```

If the user then terminates this order by pressing NEXT FORM in ENTRY, the value of STOTNET is transferred to the TOTNET field of the ORDER\_TOTALS form. Now, suppose the user goes to Browse mode *before* pressing ENTER on ORDER\_TOTALS, and changes the QTY value in the second order line. The following values are recorded:

```
(STOTNET = 123.00)
QTY = 5  UPRICE = 15.00  TPRICE = 75.00    (STOTNET = 198.00)
```

As you can see, the value of STOTNET is now grossly incorrect. Instead of adding a net change of \$15.00 to the total order, \$75.00 has been added.

If you make a similar change to an order line *after* pressing ENTER on the ORDER\_TOTALS form, the values of STOTNET and TOTNET are not easily predicted. The value of STOTNET is reset for the next form, and is either zero or contains some other value if new totals have been calculated. The value of TOTNET remains set to the last value moved to it from STOTNET.

```

-
*****ABC MANUFACTURING*****
Order Number: [100001]           Date: [09/02/86 ]

Total Net Price: [ 52.50]
Discount % [10]           Discount: [ ]
Subtotal: [ ]
Tax Rate [6.25]           Tax: [ ]
Shipping: [.00]
TOTAL: [ ]

Customer Name [ ]

ENTRY v.uu.ff           Batch Record #2           Mode: Collect
HEAD                   PRINT REFRESH                   NEXT           BROWSE           EXIT
FORM

```

**Figure 7-14. The Field Menu for TPRICE**

So what is the solution? For this particular set of forms, you can change the specifications on TPRICE so that STOTNET (and consequently TOTNET) reflects changes to the order line as long as they are made while the user is still entering data for the current order. Once the user presses ENTER to record the totals on ORDER\_TOTALS, and then makes changes to a previously entered order, the user must recalculate the value of TOTNET manually.

**Action**

As a first step, go to the Field Menu for TPRICE and enter the specifications shown in Figure 7-14.

These statements insure that any changes made to an order in Browse mode are correctly accumulated into STOTNET and correctly transmitted to TOTNET, *but only* if the user has not pressed ENTER and is still working in the current order.

---

**NOTE** A point to remember is that ENTRY is designed for applications in general. Intrinsic are provided with VPLUS/V that can be used by your data processing staff to modify ENTRY so that it is tailored to your particular application.

---

```

-
*****ABC MANUFACTURING*****
Order Number: [100001] Date: [09/02/86 ]

SHIP TO:
Name [RANDOLPH BROTHERS ]
Address [14327 University Place ]
City [Santa Clara ] State [Ca]
Zip [95050 ] Telephone [(408)327-6600 ]

QTY PART NO. DESCRIPTION UNIT TOTAL
PRICE PRICE
5 PA123-abc 10.50 52.50

ENTRY v.uu.ff Batch Record #1 Mode: Browse
FIRST DELETE PRINT REFRESH PREV NEXT COLLECT EXIT
REC REC REC REC REC REC REC REC

```

**Figure 7-15. The Field Menu for ORDNUM**

One more change is required to the ORDER\_TOTALS form to insure correct data in this situation. When NEXT FORM is pressed to stop the repeat of ORDER\_LINE, the next form, ORDER\_TOTALS, is displayed. Whenever this form is displayed in Collect mode, all the initialization procedures are performed. One of these procedures sets the order number from the SORDNUM save field and then increments the value of SORDNUM by 1. After the user goes to Browse mode to correct an order line and then returns to Collect mode, the ORDER\_TOTALS form is displayed again. The initialization procedures are performed again, which means that the save field SORDNUM is incremented again. This causes the order number on the ORDER\_TOTALS form to be greater than the order number for the same order on ORDER\_HEAD.

**Action**

Because of this, you change the field type from D for Display-Only to R for required on the Field Menu for ORDNUM as shown in Figure 7-15. Press ENTER.

Changing the field type to required allows the user to change the order number if it is greater than the correct number for the current order.

Changes cannot be made after an order is completed because the user cannot change the corresponding value of TOTNET in ORDER\_TOTALS, which is a display-only field protected from user entry. This field would also need to be changed to a required field to allow the user to correct the total if necessary, but for simplicity in this exercise, leave TOTNET as

display-only.

## TOPIC: CHANGE FORM SEQUENCE PROGRAMMATICALLY

The basic control of forms sequence is established on the Form Menu. For example, the repeating form ORDER\_LINE is defined on its Form Menu as a repeating appended form by the A in the Repeat Option selection box; and the frozen first form ORDER\_HEAD is so defined by an F in the Next Form selection box. A CHANGE statement is provided by FORMSPEC that allows you to change these specifications during execution of the forms file.

### CHANGE Statement

The CHANGE statement allows you to change form specifications as follows:

The current form can be changed to:

- NOREPEAT
- REPEAT
- REPEAT APPEND

The next form option can be changed to:

- CLEAR
- APPEND
- FREEZE APPEND

and the next form name can be changed to:

- "form name"
- field name (field contains a form name)
- index retrieve (select from form names)

- \$RETURN
- \$HEAD
- \$END
- \$REFRESH

Usually, the CHANGE statement is used in conjunction with an IF statement so that the form option or next form name is changed only under specified circumstances.

At this point in the exercise, there is no limit to the number of times the repeating form ORDER\_LINE is issued. If the form is appended so many times that it fills the screen, the first ORDER\_LINE is rolled off the screen, leaving ORDER\_HEAD frozen on the screen above the second ORDER\_LINE. This continues until the user presses NEXT FORM in ENTRY to stop the repeat and display the next different form.

Now let's use the CHANGE statement to stop the repeat automatically when the screen is full. In other words, in the exercise, you will use the CHANGE statement to change the current form to NOREPEAT after it has been repeated a specified number of times. This forces the repeat to terminate and the next different form to be displayed even if the user does not press NEXT FORM in ENTRY. As another example, you will also use

the CHANGE statement to terminate the display of forms before the order number reaches the value 1,000,000. (As presently defined, the order number is reset to 100,001 when it reaches 999,999.)

**Action**

For a full discussion of the CHANGE statement, refer to Section 4 of the VPLUS/V Reference Manual.

```
-
                               *****ABC MANUFACTURING*****
Order Number: [100001]                Date: [09/02/86 ]

                               Total Net Price: [ 52.50]
Discount % [10]                    Discount: [ .00]
                               Subtotal: [ 52.50]
Tax Rate [6.25]                    Tax: [ 3.28]
                               Shipping: [ .00]
                               TOTAL: [ 55.78]

Customer Name [RANDOLPH BROTHERS ]

ENTRY v.uu.ff      Batch Record #2      Mode: Browse
FIRST  | DELETE  | PRINT  | REFRESH  |      | PREV  | NEXT  | COLLECT  | EXIT
REC    | REC    |       |          |      | REC  | REC  |          |

```

**Figure 7-16. The LINE\_COUNT Save Field**

The first step is to change the repeat option of ORDER\_LINE so that it is terminated automatically when ORDER\_LINE has been repeated seven times. (ORDER\_LINE can be repeated seven times before the screen is full.) You need to use a save field as the counter to determine the number of times ORDER\_LINE is repeated; this save field is initialized by a field in ORDER\_HEAD, and incremented in the initialization phase of a field in ORDER\_LINE.

**Action**

Press MAIN/RESUME to display the Main Menu; use option S--add a save field; press ENTER.

On the Save Field Menu, name the save field LINE\_COUNT and specify that it is data type of DIG with a length of two digits as shown in Figure 7-16. Press ENTER.

\*\*\*\*\*ABC MANUFACTURING\*\*\*\*\*

Order Number: [100002]

Date: [09/02/86]

SHIP TO:

Name [BOND FABRICS, INC.]  
Address [5570 15th Street]  
City [San Jose] State [Ca]  
Zip [95162] Telephone [(408)334-7790]

QTY	PART NO.	DESCRIPTION	UNIT PRICE	TOTAL PRICE
100	PA123-xh1		10.50	

ENTRY v.uu.ff      Batch Record #3      Mode: Collect  
HEAD      PRINT      REFRESH      NEXT      BROWSE      EXIT  
FORM

Figure 7-17. The Field Menu for QTY

Since QTY is the first field in ORDER\_LINE, you use that field to increment the LINE\_COUNT save field and conditionally terminate the repeat.

**Action**

Return to the Main Menu and use option G--to go to the QTY field of ORDER\_LINE, where you specify the statements to increment the LINE\_COUNT save field, as shown in Figure 7-17.

Note that the existing edit statement must be preceded by a FIELD heading so that it is not executed in the initialization phase. Since the other statements are executed in the initialization phase, they are performed before the form is displayed on the screen. Now you need to give LINE\_COUNT an initial value because it must be initialized each time a new order transaction is started.

**Action**

Return to the Main Menu and use option G--to go to the ORDNUM field of ORDER\_HEAD, where you specify the statements: INIT and SET LINE\_COUNT TO 0.

These statements initialize the LINE\_COUNT save field to zero. The counter can be incremented in any field in ORDER\_LINE, but it must be incremented in the initialization phase so that the count is not affect-

ed by modifications made in Browse mode.

```

-
*****ABC MANUFACTURING*****
Order Number: [100002]           Date: [09/02/86 ]
                                Total Net Price: [ 1050.00]
Discount % [10]                 Discount: [ ]
                                Subtotal: [ ]
Tax Rate [6.25]                 Tax: [ ]
                                Shipping: [.00 ]
                                TOTAL: [ ]
Customer Name [ ]
ENTRY v.uu.ff      Batch Record #4      Mode: Collect
HEAD              PRINT REFRESH          NEXT      BROWSE  EXIT
FORM              FORM                   FORM

```

Figure 7-18. The Field Menu for ORDNUM

The next example of using a CHANGE statement affects the Next Form option. The current specification for the ORDNUM field has a length of six digits. Thus, if the order number is incremented above the value 999999, it no longer fits in the field. As currently defined, the order number is reset to 100001 before it overflows the field. Another solution is to terminate execution of the forms file when the order number gets too big.

**Action**

Change the Field Menu of ORDNUM in the ORDER\_TOTALS form as shown in Figure 7-18. Press ENTER.

This completes the specifications for the form ORDFORM1. Table 7-1 summarizes the global specifications for ORDFORM1, including the four save fields; Table 7-2, Table 7-3 and Table 7-4 summarizes the field specifications for all the forms in the file.

**Action**

Return to the Main Menu and compile the form so that it can be used for data entry. Press EXIT.

Table 7-1. Summary of Globals and Save Fields in ORDFORM1

Global Specifications				
Head Form	Enhancements		Window	
	Field	Error	Line No.	Enhancement
ORDER_HEAD	HI	IU	1	HI
Save Fields				
Name	Length	Data Type	Initial Value	
SORDNUM	6	DIG	100001	
STOTNET	10	NUM2	0	
SNAME	42	CHAR	--	
LINE_COUNT	2	DIG	0	

Table 7-2. Summary of ORDFORM1 Fields

ORDER_HEAD Repeat Option = N Next Form = F Next Form Name = ORDER_LINE					
Field		FType	DType	Processing Specifications	Initial Value
Num	Name				
0	ORDNUM	D	DIG	INIT SET ORDNUM TO SORDNUM SET STOTNET TO 0 SET LINE_COUNT TO 0	
1	ORDATE	D	MDY	INIT SET ORDATE TO \$TODAY	
2	NAME	R	CHAR	SET NAME TO SNAME	
3	ADDRESS	O	CHAR		
4	CITY	O	CHAR		
5	ST	O	CHAR	IN \$STATE	
6	ZIP	O	CHAR	MATCH dddd[-ddd]	
7	PHONE	O	CHAR	MATCH [(ddd)]b*ddd-ddd	

Table 7-3. Summary of ORDFORM1 Fields (cont'd)

ORDER_LINE Repeat Option = A Next Form = C Next Form Name = ORDER_TOTALS					
Field		FType	DType	Processing Specifications	Initial Value
Num	Name				
9	QTY	D	DIG	INIT SET LINE COUNT TO LINE COUNT + 1 IF LINE_COUNT GE ? THEN CHANGE CFORM TO NOREPEAT FIELD GE 1 "Enter quantity equal to or greater than 1"	
10	PARTNUM	O	CHAR	MATCH Pudd-???	
11	DESCRIPTION	O	CHAR		
12	UPRICE	O	NUM2	JUSTIFY RIGHT STRIP LEADING "0"	
13	TPRICE	D	NUM2	FIELD IF EQ \$EMPTY THEN SET TPRICE TO QTY * UPRICE SET STOTNET TO STOTNET + TPRICE ELSE SET STOTNET TO STOTNET TPRICE SET TPRICE TO QTY * UPRICE SET STOTNET TO STOTNET + TPRICE	

Table 7-4. Summary of ORDFORM1 Fields (cont'd)

ORDER_TOTALS Repeat Option = N Next Form = C Next Form Name = \$HEAD					
Field		FType	DType	Processing Specifications	Initial Value
Num	Name				
1	ORDNUM	D	DIG	INIT SET ORDNUM TO SORDNUM IF GT 999999 THEN CHANGE NFORM TO \$END ELSE SET SORDNUM TO SORDNUM + 1	
2	ORDATE	D	MDY	INIT SET ORDATE TO \$TODAY	
3	TOTNET	D	NUM2	INIT SET TOTNET TO \$TOTNET FIELD SET TOTNET TO \$TOTNET	
4	DISCNT_PRCNT	O	NUM0	--	10
5	DISCNT	D	NUM2	IF TOTNET GE 1000 THEN SET DISCNT TO DISCNT_PRCNT % TOTNET ELSE SET DISCNT TO 0	
6	SUB	D	NUM2	SET SUB TO TOTNET - DISCNT	
7	TAX_RATE	R	NUM2	--	6.25
8	TAX	D	NUM2	SET TAX TO TAX_RATE % SUB	
9	SHIP	O	NUM2	JUSTIFY RIGHT STRIP LEADING "0"	.00
10	TOTAL	D	NUM2	SET TOTAL TO SUB + TAX + SHIP	
11	NAME	D	CHAR	SET NAME TO \$NAME	

:RUN\_ENTRY.PUB.SYS

HP32209v.uu.ff ENTRY (C) HEWLETT-PACKARD CO. 1986

ENTER FORMS FILE NAME AND PRESS RETURN: ORDFORM1

ENTER BATCH FILE NAME AND PRESS RETURN: DATA7

\*\*\*\*\*ABC MANUFACTURING\*\*\*\*\*

Order Number: [100001]

Date: [09/02/86]

SHIP TO:

Name	[		]
Address	[		]
City	[		]
Zip	[		]
Telephone	[		]
State	[		]

QTY	PART NO.	DESCRIPTION	UNIT PRICE	TOTAL PRICE
-----	----------	-------------	------------	-------------

HEAD FORM		PRINT	REFRESH		NEXT FORM	BROWSE	EXIT
-----------	--	-------	---------	--	-----------	--------	------

Figure 7-19. The ORDER\_HEAD Form as Initially Displayed

**ENTER DATA ON REPEATING FORM**

Now that ORDFORM1 has been redefined, run ENTRY and enter data into a new batch file.

**Action**

Issue the run command as shown and specify the forms file and batch file names, ORDFORM1 and DATA7.

ENTRY displays the first form, ORDER\_HEAD, as shown in Figure 7-19. Note that the window line is now positioned at the top of the form and that the line for the order information is not displayed yet.

\*\*\*\*\*ABC MANUFACTURING\*\*\*\*\*

Order Number: [100001]

Date: [09/02/86]

SHIP TO:

Name [MURPHY BROTHERS ]  
Address [3260 Main Street ]  
City [Sunnyvale ] State [Ca]  
Zip [95156 ] Telephone [(408) 246-3300]

QTY	PART NO.	DESCRIPTION	UNIT PRICE	TOTAL PRICE

Figure 7-20. The ORDER\_HEAD Form Frozen with the ORDER\_LINE

**Form Appended**

The only data you are required to enter on this form is the customer name. However, for most purposes you also enter the customer's address and phone number.

**Action**

Type in the customer information on the header form as shown in Figure 7-20 and press ENTER to record this information in the first record of the batch file.

Because the header form is defined as a "frozen" form, it remains on the screen when you press ENTER. The next form, which is the line of order information, is a repeating appending form and is displayed on the screen

immediately below the first form. The fields on this next form are aligned with the field headers at the bottom of the first form.

Note that the record number in the window line is now shown as #2. The data you enter in the new form is recorded as the second record of the DATA7 batch file. Unless you go to Browse mode, you cannot change the data in the header form; the entire screen except for the current form (ORDER\_LINE) is now protected.

ENTRY v.uu.ff Batch Record #2 Mode: Collect

\*\*\*\*\*ABC MANUFACTURING\*\*\*\*\*

Order Number: [100001]

Date: [09/02/86]

SHIP TO:

Name [MURPHY BROTHERS ]  
Address [3260 Main Street ]  
City [Sunnyvale ] State [Ca]  
Zip [95156 ] Telephone [(408) 246-3300]

QTY	PART NO.	DESCRIPTION	UNIT PRICE	TOTAL PRICE
2	PX111-ccc		15.00	

HEAD FORM    PRINT    REFRESH    NEXT FORM    BROWSE    EXIT

Figure 7-21. Record No. 2 with Data

**Action**

Type data into all the fields on this form except the total price field, as shown in Figure 7-21; press ENTER.

\*\*\*\*\*ABC MANUFACTURING\*\*\*\*\*

Order Number: [100001]

Date: [09/02/86]

SHIP TO:

Name [MURPHY BROTHERS]
Address [3260 Main Street]
City [Sunnyvale] State [Ca]
Zip [95155] Telephone [(408) 246-3300]

Table with 5 columns: QTY, PART NO., DESCRIPTION, UNIT PRICE, TOTAL PRICE. Row 1: 2, PX111-ccc, [redacted], 15.00, 30.00. Row 2: 5, PB345-XYZ, [redacted], 20.00, [redacted].

HEAD FORM PRINT REFRESH NEXT FORM BROWSE EXIT

Figure 7-22. The ORDER\_LINE Form Appended for Record No. 3

When you press ENTER to record the data in the first order line, the unit price is formatted, the total price is calculated and displayed, and then a new ORDER\_LINE form is repeated immediately below the last form.

Action

Now type data into the second order line, as shown in Figure 7-22. Press ENTER.

\*\*\*\*\*ABC MANUFACTURING\*\*\*\*\*

Order Number: [100001]

Date: [09/02/86]

SHIP TO:

Name [MURPHY BROTHERS ]  
Address [3260 Main Street ]  
City [Sunnyvale ] State [Ca]  
Zip [95156 ] Telephone [(408) 246-3300]

QTY	PART NO.	DESCRIPTION	UNIT PRICE	TOTAL PRICE
2	PX111-ccc		15.00	30.00
5	PB345-XYZ		20.00	100.00

HEAD FORM [ ] PRINT REFRESH [ ] NEXT FORM BROWSE EXIT

Figure 7-23. The ORDER\_LINE Form Appended for Record No. 4

When you press ENTER, the second line of order information is recorded as Record #3 of the batch file. The total price is displayed and a new blank order line form for Record #4 appears directly below the last line, as shown in Figure 7-23.

\*\*\*\*\*ABC MANUFACTURING\*\*\*\*\*

Order Number: [100001] Date: [09/02/86]

Total Net Price: [ 130.00]  
Discount % [10] Discount: [ ]  
Subtotal: [ ]  
Tax Rate [6.25] Tax: [ ]  
Shipping: [.00]  
TOTAL: [ ]

Customer Name [ ]

HEAD FORM [ ] PRINT REFRESH [ ] NEXT FORM BROWSE EXIT

**Figure 7-24. The ORDER\_TOTALS Form as Initially Displayed**  
**TERMINATE REPEATING FORM**

Suppose there is no more order information and you want to record the totals for the current order.

**Action**

Press NEXT FORM.

The screen is cleared and the final form, ORDER\_TOTALS, displayed on the screen. The totals from individual order lines have been summed to determine the net total for the order, which is displayed on this screen along with the order number and date, as shown in Figure 7-24.

Although the form design would allow the user to change the tax rate, the discount percentage, or enter a shipping charge, it is not necessary.

**Action**

Do not enter data on this form; press ENTER to record the totals on batch Record #4.

ENTRY v.uu.ff Batch Record #5 Mode: Collect

\*\*\*\*\*ABC MANUFACTURING\*\*\*\*\*

Order Number: [100002]

Date: [09/02/86]

SHIP TO:

Name [ ]  
Address [ ]  
City [ ] State [ ]  
Zip [ ] Telephone [ ]

QTY	PART NO.	DESCRIPTION	UNIT PRICE	TOTAL PRICE
-----	----------	-------------	------------	-------------

HEAD FORM [ ] PRINT REFRESH [ ] NEXT FORM BROWSE EXIT

Figure 7-25. The ORDER\_HEAD Form

When you press ENTER, the screen is cleared and a new header form is displayed, as shown in Figure 7-25. The new form has the order number 100002 since it is the start of a new order.

\*\*\*\*\*ABC MANUFACTURING\*\*\*\*\*

Order Number: [100002] Date: [09/02/86]

SHIP TO:  
 Name [COLLINS FURNITURE]  
 Address [5001 Mary Avenue]  
 City [Sunnyvale] State [Ca]  
 Zip [95166] Telephone [(405) 247-9501]

QTY	PART NO.	DESCRIPTION	UNIT PRICE	TOTAL PRICE
3	PH222-756	3-tier in/out basket	15.00	45.00
1	PB123-900	Desk chair	150.00	150.00
5	PB125-90a	Adjustable stool	75.00	375.00
10	PC122-abc	5-hole punch	5.00	50.00
4	PX101-bcc	pencil sharpener	6.00	24.00
7	PJ110-999	stapler	3.00	21.00
15	PK113-909	typewriter stand	50.00	750.00
HEAD FORM		PRINT REFRESH	NEXT FORM	BROWSE EXIT

Figure 7-26. The ORDER\_LINE Form Appended the Maximum Number of Times

**AUTOMATIC TERMINATION OF REPEATING FORM**

In the order just entered, the repeating form was terminated by pressing NEXT FORM. It is also terminated automatically after seven lines have been entered on a single order. This termination is solely a result of how the forms file ORDFORM1 was designed. It could have been designed so that the repeat stopped after 10 forms, after 30 forms, or by default, only when NEXT FORM is pressed.

In any case, because of the forms file design, the final order form ORDER\_TOTALS is displayed automatically after the seventh order line is entered.

**Action**

Enter the data shown in Figure 7-26 into the batch file.

When you press ENTER to record the seventh (and last) order line on the screen, the screen is cleared automatically and the next form is displayed.

\*\*\*\*\*ABC MANUFACTURING\*\*\*\*\*

Order Number: [100002] Date: [09/02/86]

Total Net Price: [ 1415.00]  
Discount % [10] Discount: [ ]  
Subtotal: [ ]  
Tax Rate [6.25] Tax: [ ]  
Shipping: [.00]  
TOTAL: [ ]

Customer Name [ ]

HEAD FORM PRINT REFRESH NEXT FORM BROWSE EXIT

Figure 7-27. The ORDER\_TOTALS Form

When you press ENTER to record the last line of data for the repeating form, the total for that line is calculated. Then the screen is cleared and the next form is displayed. The accumulated totals for all the order lines are summed into the net total and displayed on the last form of the order, as shown in Figure 7-27.

\*\*\*\*\*ABC MANUFACTURING\*\*\*\*\*

Order Number: [100002] Date: [09/02/86]

	Total Net Price:	[ 1415.00]
Discount % [10]	Discount:	[ 141.50]
	Subtotal:	[ 1273.50]
Tax Rate [6.25]	Tax:	[ 79.59]
	Shipping:	[ .00]
	TOTAL:	[ 1353.09]

Customer Name [COLLINS FURNITURE]

FIRST REC	DELETE REC	PRINT	REFRESH	PREV REC	NEXT REC	COLLECT	EXIT
-----------	------------	-------	---------	----------	----------	---------	------

Figure 7-28. Browse Mode, Record No. 13

**Action**

Press ENTER to have ENTRY calculate and record the remaining totals as Record #13 of the batch file. Then press BROWSE and the record appears as shown in Figure 7-28.

**Modify Accumulated Values**

Usually, modification of existing records is a simple matter--you locate the record you want to change, make the change, and then press ENTER to record it. If the corrected value affects a value in another record, you locate that record and press ENTER to record the new value (as described in Section 6). When totals are accumulated, making changes is more involved, particularly if the summed values are entered on a repeating form.

As the forms in ORDFORM1 are designed, you can change a value in the repeating order line that affects the line totals and have this value correctly summed into the net total field only if you make the change *before* you press ENTER to record the totals. If you wait until the totals have been entered in the last record of an order, the net total will probably be incorrect. If you must change either the quantity or unit price on an order line *after* the order totals are recorded, you would then have to correct the net total value yourself. However, the net total field on the totals form does not allow user entry. Thus, the solution is to delete all the records in an order and then enter the order correctly, as described later in this section.

\*\*\*\*\*ABC MANUFACTURING\*\*\*\*\*

Order Number: [100001]

Date: [09/02/86]

SHIP TO:

Name [MURPHY BROTHERS]  
 Address [3260 Main Street]  
 City [Sunnyvale] State [Ca]  
 Zip [95156] Telephone [(408) 246-3300]

QTY	PART NO.	DESCRIPTION	UNIT PRICE	TOTAL PRICE
-----	----------	-------------	------------	-------------

FIRST REC	DELETE REC	PRINT	REFRESH	PREV REC	NEXT REC	COLLECT	EXIT
-----------	------------	-------	---------	----------	----------	---------	------

Figure 7-29. Browse Mode, Record No. 1

**DELETE RECORDS FROM BATCH FILE**

To delete a record from the batch file, you must be in Browse mode. Then, you locate the record to be deleted and press DELETE REC. The record is effectively deleted from the batch file and can no longer be referenced. This means if you try to view it in Browse mode, it is skipped. Furthermore, the space in the file that was occupied by the deleted records is no longer available. You cannot delete a record and then insert a new record in that location; all new records are added to the end of the batch file.

In our example, anywhere from three to nine records may comprise a single order depending on the number of records generated from the repeating form. Thus, if you want to delete an order, you must delete all the records in the order. You can, of course, delete single records from an order, but if the deleted record affects the total price, you must then manually change this total on the last form of the order and then press ENTER to record the changed total.

Since in most cases you want to delete an entire order rather than a single record in that order, this exercise gives an example that deletes order number 100001, the first order in the current batch file.

## Action

To delete this order, first go to Browse mode by pressing BROWSE. Then press FIRST REC to find the first record in the file. In this case, it is also the first record of the order to be deleted. The record appears as shown in Figure 7-29.

```
ENTRY v.uu.ff          Batch Record #2          Mode: Browse
2  PX111-ccc          17.50          35.00
```

```
FIRST REC  DELETE REC  PRINT  REFRESH          PREV REC  NEXT REC  COLLECT  EXIT
```

Figure 7-30. Browse Mode, Record No. 2

## Action

When you press DELETE REC, the screen is cleared and the next record in sequence is displayed, as shown in Figure 7-30. Since Record #2 is part of order number 100001, press DELETE REC to delete it.

```
ENTRY v.uu.ff          Batch Record #3          Mode: Browse
5  PB345-XYZ          20.00  100.00
```

```
FIRST  DELETE  PRINT  REFRESH          PREV  NEXT  COLLECT  EXIT
REC    REC                                REC    REC
```

**Figure 7-31. Browse Mode, Record No. 3**

When it is deleted, it is replaced by the next record in sequence, as shown in Figure 7-31.

**Action**

Delete this record as well.

\*\*\*\*\*ABC MANUFACTURING\*\*\*\*\*

Order Number: [100001] Date: [09/02/86]

	Total Net Price:	[ 135.00]
Discount % [10]	Discount:	[ .00]
	Subtotal:	[ 135.00]
Tax Rate [6.25]	Tax:	[ 8.44]
	Shipping:	[ .00]
	TOTAL:	[ 143.44]

Customer Name [MURPHY BROTHERS]

FIRST REC	DELETE REC	PRINT	REFRESH	PREV REC	NEXT REC	COLLECT	EXIT
--------------	---------------	-------	---------	-------------	-------------	---------	------

Figure 7-32. Browse Mode, Record No. 4

The next record is then displayed, as shown in Figure 7-32.

**Action**

Delete Record #4 in the same way.

When you delete Record #4, you have deleted the entire order number 100001. Should you wish to enter another order for this company, it is added to the end of the existing orders with new record numbers and a new order number.

**Action**

The exercise is complete; press EXIT.

## Chapter 8 Document Forms and Compile Fast Forms File

The sample forms file used in this exercise, called ORDFORM1, has now been completely defined and is ready to use in production mode.

In this last section you list the entire forms file to the line printer so that you have a formal definition of the file contents. But, before you list it, you make one last set of changes to the file. By adding comments to the field processing specifications in each field, you enable

forms designers responsible for maintenance of this file to understand the specification logic. These comments appear both on the line printer listing and when the Field Menus are displayed at the terminal; they are ignored when the file is executed and do not change how the file operates.

Finally, you compile the forms file to a new type of file, called a **fast forms file**, that protects the forms from modification and may make the execution of the file more efficient.

ENTRY also prints forms to a line printer, with or without data, in order to have a listing that shows how the forms actually look on the screen. Using ENTRY, you print an image of each form on the line printer. Then, after entering data, you print the form with its data for each record of a single order. This section illustrates the following features:

Using FORMSPEC:

\* Add comments to processing specifications

(Field Menu).

\* List entire forms file or individual forms

(Main Menu).

\* Compile to fast forms file (Main Menu).

Using ENTRY:

\* Use PRINT to print forms or records.

---

```
| :RUN FORMSPEC.PUB.SYS |
```

---

FORMSPEC v.uu.ff Forms File Menu

Forms File Name [ORDFORM1]

-

PREV FORM	NEXT FORM		REFRESH		PREV	NEXT	MAIN/ RESUME	EXIT
--------------	--------------	--	---------	--	------	------	-----------------	------

Figure 8-1. Run FORMSPEC

#### ADD COMMENTS TO PROCESSING SPECIFICATIONS

When you entered the editing statements in Section 4, you added custom error messages that are displayed by ENTRY in case the edit fails. These messages are aids to the user to make error correction easier. FORMSPEC provides another type of message for the forms designer. These are comments that can be added to any processing specification to describe the purpose of the specification statement. These comments are particularly useful for non-editing statements where custom error messages are not allowed.

Comments must be preceded by a back slash (\). This distinguishes them from the specification statement itself. When the forms file is compiled, the comments are retained in the forms file, but they do not affect execution and, thus, are governed by no syntax rules other than the initial back slash.

When the Field Menu is displayed, the comment is displayed too. And when the forms file is listed, the comments are listed with the field description. They serve as documentation for the forms designer to describe what the particular statement does and, as such, are particularly useful if someone other than the initial designer wants to determine the intent of the associated statement.

#### Action

To add these statements, you first run FORMSPEC and request the file,

ORDFORM1, as shown in Figure 8-1.

```
FORMSPEC v.uu.ff Main Menu                                FORMS FILE: ORDFORM1
[G] Enter Selection                                     Forms File is 3% full
A--Add a form
S--Add a Save field
T--Terminal/Language Selection Menu
G--Go to GLOBALS Menu, OR Go to form [ORDER_TOTALS ] field [ORDNUM ]
L--List Forms File, OR List form ... [ ]
D--Delete Save field ..... [ ]
   Form ..... [ ]
C--Copy new form name ..... [ ]
   from form ..... [ ]
   from Forms File (opt) ..... [ ]
X--Compile Forms File
   Optional: Fast Forms File [ ]
R--Relate child form [ ] to parent form [ ]

PREV FORM   NEXT FORM   REFRESH   PREV   NEXT   MAIN/RESUME   EXIT
```

Figure 8-2. The Main Menu

**Action**

Then, when the Main Menu is displayed you go to the Field Menu for ORDNUM in the ORDER\_TOTALS form, as shown in Figure 8-2. Press ENTER to go to the specified field.

```

FORMSPEC v.uu.ff Field Menu                                FORM: ORDER_TOTALS
*****ABC MANUFACTURING*****

Order Number: [ordnum]                                     Date: [ordate] ]
      ^
Num [1 ] Len [6 ] Name [ORDNUM ] Enh [HI ] FType [D] DType [DIG ]
Initial Value [ ]
-----
*** Processing Specifications ***
INIT
SET TO SORDNUM                                           \Move order number from save field
IF SORDNUM GT 999999 THEN
  CHANGE NFORM TO $END                                   \Terminate before 1,000,000 orders
ELSE SET SORDNUM TO SORDNUM + 1                          \Otherwise, increment order number

PREV  NEXT  FIELD  REFRESH                                PREV  NEXT  MAIN/  EXIT
FORM  FORM  TOGGLE                                     FORM  RESUME

```

Figure 8-3. The Field Menu for ORDNUM in ORDER\_TOTALS Form

**Action**

When the Field Menu for ORDNUM in the ORDER\_TOTALS form is displayed, enter the comments shown in Figure 8-3; press ENTER.

Comments do not need to be added to editing specification statements since these all have custom error messages that sufficiently explain the statement without further comment.

**Action**

Now that you understand the sequence of entering comments, look at the listing of the forms file in Appendix A and enter the comments for each Field Menu.

```

FORMSPEC v.uu.ff Main Menu                                FORMS FILE: ORDFORM1
[L] Enter Selection
A--Add a form
S--Add a Save field
T--Terminal/Language Selection Menu
G--Go to GLOBALS Menu, OR Go to form [ ] field [ ]

L--List Forms File, OR List form ... [ ]

D--Delete Save field ..... [ ]
   Form ..... [ ]

C--Copy new form name ..... [ ]
   from form ..... [ ]
   from Forms File (opt) ..... [ ]

X--Compile Forms File
   Optional: Fast Forms File [ ]

R--Relate child form [ ] to parent form [ ]

PREV  NEXT  REFRESH  PREV  NEXT  MAIN/  EXIT
FORM  FORM  FORM      FORM  FORM  RESUME

```

Figure 8-4. The Main Menu

**LIST ENTIRE FORMS FILE**

Assuming you have run FORMSPEC and opened a forms file, the steps to get a listing of the forms file are:

1. On the Main Menu, specify a L in the selection box and
2. Press ENTER.

**Action**

Press MAIN/RESUME to go to the Main Menu. Using the steps outlined above, list the ORDFORM1 forms file. Figure 8-4 shows the value to be entered.

The listing of forms file ORDFORM1 is shown in Appendix A.

[L] Enter Selection

```

A--Add a form
S--Add a Save field
T--Terminal/Language Selection Menu
G--Go to GLOBALS Menu, OR Go to form [          ] field [          ]

L--List Forms File, OR List form ... [ORDER_HEAD]

D--Delete Save field ..... [          ]
   Form ..... [          ]

C--Copy new form name ..... [          ]
   from form ..... [          ]
   from Forms File (opt) ..... [          ]

X--Compile Forms File
   Optional: Fast Forms File [          ]

R--Relate child form [          ] to parent form [          ]

```

PREV FORM	NEXT FORM		REFRESH		PREV	NEXT	MAIN/ RESUME	EXIT
--------------	--------------	--	---------	--	------	------	-----------------	------

Figure 8-5. The Main Menu

### LIST SINGLE FORM

Assuming you have run FORMSPEC and opened a forms file, the steps to list a single form are:

1. On the Main Menu, specify a L in the selection box and the name of the form to be listed.
2. Press ENTER.

### Action

Using the steps outlined above, list the ORDER\_HEAD form. Figure 8-5 shows the values to enter on the Main Menu.

The listing of a single form is identical to that of each form on the forms file listing. The listing of the ORDER\_HEAD form is shown as part of Appendix A.

### LIST TO PARTICULAR DEVICE

When you request a form or forms file listing on the Main Menu of FORMSPEC, the listing is automatically printed on a line printer. If you want to specify a particular line printer, you must use a file equation (MPE :FILE command).

The formal designator of the FORMSPEC list file is FORMLIST. This is the name you should specify in the file equation. For example, to direct a listing to an HP 2680 laser printer whose device name is PP, you would need to exit FORMSPEC to return to MPE and enter the following command:

```
:FILE FORMLIST;DEV=PP
```

Do not try this yet because the forms file has not been compiled.

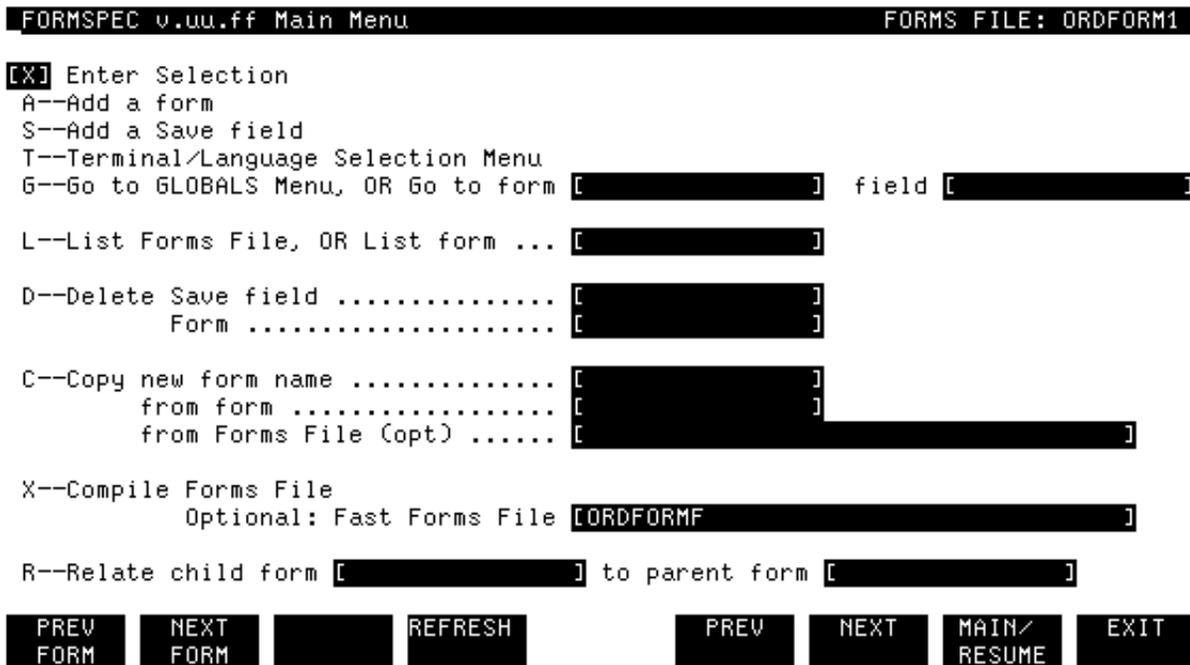


Figure 8-6. The Main Menu

#### COMPILE TO FAST FORMS FILE

After your forms file is completely defined so that you expect no further modifications, you can compile the file to a fast version. A fast forms file uses only the compiled machine-readable version of the forms definitions, thereby using fewer records. It also reduces the record size to a minimum. As a result, this condensed forms file can be executed significantly faster than a regular forms file.

A fast forms file is not, however, acceptable to FORMSPEC. Thus, it cannot be modified. If you must modify your forms file after compiling it to a fast forms file, you should use your original forms file and then recompile it to the fast file.

Assuming you have run FORMSPEC and opened a forms file, the steps to compile a forms file to a fast forms file are:

1. On the Main Menu, specify a X in the selection box and a name for

the fast forms file.

2. Press ENTER.

### Action

Using the steps outlined above, compile the ORDFORM1 forms file to a fast forms file called ORDFORMF. Figure 8-6 shows the values to enter on the Main Menu.

When you press ENTER, the fast forms file is compiled. Although compilation to this file may take longer than a normal compilation, subsequent execution by ENTRY is faster.

```
:RUN ENTRY.PUB.SYS  
HP32209v.uu.ff ENTRY (C) HEWLETT-PACKARD CO. 1986  
ENTER FORMS FILE NAME AND PRESS RETURN: ORDFORMF  
ENTER BATCH FILE NAME AND PRESS RETURN: DATA7
```

```
ENTRY v.uu.ff          Batch Record #20          Mode: Collect  
  
*****ABC MANUFACTURING*****  
Order Number: [100004]          Date: [09/02/86 ]  
  
SHIP TO:  
Name [                ]  
Address [                ]  
City [                ] State [  ]  
Zip [                ] Telephone [                ]  
  
QTY   PART NO.   DESCRIPTION                               UNIT   TOTAL  
                                           PRICE  PRICE  
  
[HEAD] [ ] [PRINT] [REFRESH] [ ] [NEXT] [BROWSE] [EXIT]  
[FORM] [ ] [ ] [ ] [FORM] [ ] [ ]
```

Figure 8-7. Run ENTRY--The Next Sequential Record

### RUN ENTRY WITH FAST FORMS FILE

Since the forms file was recompiled (to a fast forms file), you will need to respond to a warning message when you now run ENTRY.



The form is printed on the line printer exactly as it was displayed on the screen, except that no enhancements show, as shown in Figure 8-8. A line printer has no way to indicate terminal enhancements such as blinking or half-bright. Also, the window line is not printed, nor are any values that are typed into fields but not entered. Initial values displayed with the form are, however, printed when the form is printed.

**NOTE** Each time you press PRINT, the current form (or record in Batch mode) is saved, but it is not actually printed on the line printer until you exit from ENTRY by pressing EXIT.

When you press PRINT, this form is sent to the line printer. The form itself remains on the screen so you can enter data into it.

```

ENTRY v.uu.ff           Batch Record #12           Mode: Collect

*****ABC MANUFACTURING*****

Order Number: [100004]           Date: [09/16/86]

SHIP TO:
  Name [Martin & Gilbert]
  Address [254 Mountain View Ave.]
  City [Santa Clara] State [Ca]
  Zip [95050] Telephone [(408) 249-6974]

QTY  PART NO.  DESCRIPTION  UNIT PRICE  TOTAL PRICE
3    PJ123-aaa  Desk with file drawers  195.00  585.00
7    PJ112-aab  Desk chair          75.50  528.50
1    PK111-aac  Desk-top file       75.00  19.75

[HEAD] [ ] [PRINT] [REFRESH] [ ] [NEXT] [BROWSE] [EXIT]
[FORM] [ ] [FORM] [FORM] [FORM] [FORM] [FORM]
  
```

**Figure 8-9. Example with Data**

Now, if you type data into the form and press ENTER, the next form is displayed. In this case, it is a repeating form appended to the existing form.

This particular form is peculiar in that it has no headings, and no field delimiters that are printable characters. The fields in which you enter quantity, part number, description, unit price, and total price are each delimited by non-printing characters. When the form is displayed on the screen, you can see the field boundaries because the fields are enhanced--that is, they are half-bright inverse video. But these

enhancements are not printable. Therefore, when you press PRINT to print this form, a blank line printer page is issued.

Because of the nature of this form, do not attempt to print it.

**Action**

Simply enter data (except for the totals) into the repeating form until the three-line order is completed, as shown in Figure 8-9. Press ENTER for each line of the order.

Notice that ENTRY calculates the totals when ENTER is pressed.

**Action**

After the order is complete, press NEXT to display the last form, which is shown in Figure 8-10.

```
-
                               *****ABC MANUFACTURING*****
Order Number: [100004]                               Date: [07/16/85      ]

                               Total Net Price: [ 1133.25]
Discount % [10]                Discount: [ 113.33]
                               Subtotal: [ 1019.92]
Tax Rate [6.25]                Tax: [ 63.74]
                               Shipping: [ .00]
                               TOTAL: [ 1083.66]

Customer Name: [Martin & Gilbert]

[HEAD] [ ] [PRINT] [REFRESH] [ ] [NEXT] [BROWSE] [EXIT]
[FORM] [ ] [FORM] [FORM] [FORM] [FORM] [FORM]
```

**Figure 8-10. The Totals for Order Number 4**

When this last form in the order is displayed on the screen, it contains not only the initial values for the order number and the date, but also the totals accumulated from the preceding repeating form. When you press PRINT in Collect mode the form is listed with these initial values, as shown in Figure 8-11.

```

*****ABC MANUFACTURING*****
Order Number: [100004]           Date: [09/02/86]
Total Net Price: [ 1133.25]
Discount % [10]           Discount: [ 113.33]
Subtotal: [ 1019.92]
Tax rate [6.25]           Tax: [ 63.74]
Shipping: [ 0.00]
TOTAL: [ 1083.66]
Customer Name: [Martin & Gilbert]

```

Figure 8-11. Example Listing of the TOTALS Form

**PRINT DATA (BROWSE MODE)**

After data has been entered in Collect mode, you can print the entered data as it appears on the form by going to Browse mode. Then locate the records you want to print and, as each record is displayed on the screen, press PRINT.

For example, order number 10004 contains five records: the order heading, three repeating order lines, and the order total. To print the content of this order on the line printer, go to Browse mode, use PREV REC to locate the first form in Order Number 4, then press PRINT. This causes the first record with the entered data to be printed. Then press NEXT REC to display the next record and press PRINT to print this record. Although the next record is shown appended below the first record on the screen, only the record itself is printed on the line printer. Continue displaying records, and printing each record until all records in the order are printed.

Remember that the printed records are not actually listed on the line printer until you press EXIT to terminate ENTRY.

The line printer listing prints one record per page. The listing is shown in Figure 8-12 and Figure 8-13.



0	3	PJ123-aaa	desk	195.00	585.00	0
0	7	PJ112-aab	chair	75.50	528.50	0
0	1	PK111-aac	desk-top file	19.75	19.75	0

0	*****ABC MANUFACTURING*****					0	
0	Order Number: [100004]	Date: [09/02/86				0	
0		Total Net Price: [	1133.25]			0	
0	Discount % [10]	Discount: [	113.33]			0	
0		Subtotal: [	1019.92]			0	
0	Tax rate [6.25]	Tax: [	63.74]			0	
0		Shipping: [	0.00]			0	
0		TOTAL: [	1083.66]			0	
0	Customer Name: [Martin & Gilbert					]	0

Figure 8-13. Example Listing of an Order with Five Records, cont'd

The order is now completely printed. Note that record numbers are not printed.

**Action**

The exercise is complete; press EXIT.

## Appendix A Forms File Listing

```

*****
*
*                      FORMSPEC Version B.04.12
*                      FRI, AUG 8, 1986,  2:24 PM
*
*                      ODRFORM1.PUB.W
*
*****
Forms File Status
  Modified: WED, AUG 6, 1986,  3:46 PM
  Compiled: WED, AUG 6, 1986,  3:46 PM
  Requires 2474 + 60 words (SUBTRACT 800 for fast forms file)

          Head Form: ORDER_HEAD
Default Display Enhancement: HI
          Error Enhancement: IU
          Window Display Line: 1
          Window Enhancement: HI

THERE ARE 4 SAVE FIELDS IN THIS FORMS FILE:

  Save Field: SORDNUM           Length: 6           Data Type: DIG
          Init Value: 100001

  Save Field: SNAME             Length: 42          Data Type: CHAR
          Init Value:

  Save Field: STOTNET           Length: 10          Data Type: NUM2
          Init Value: 0.00

  Save Field: LINE_COUNT        Length: 2           Data Type: DIG
          Init Value:

There are 3 forms in this forms file:

          Field Counts   Largest   Number of
          Total/Display  Field    Lines in
          Only           Number   Screen   Next Form

Form Name
ORDER_HEAD      8 / 2      16      16      ORDER_LINE
ORDER_LINE     5 / 1      26       5      ORDER_TOTALS
ORDER_TOTALS   11 / 7     11      21      $HEAD

```

**Figure A-1. Forms File Listing**

FORMSPEC VERSION B.04.12 FRI, AUG 8, 1986, 2:24 PM  
 FORMS FILE: ORDFORM1.PUB.W PAGE 2

Form: ORDER\_HEAD  
 Repeat Option: N  
  
 Next Form Option: F  
 Next Form ORDER\_LINE  
 Reproduced From:

Comments: EXECUTE THIS FORM FIRST  
 \*\*\*\*\*  
  
 \*\*\*\*\*ABC MANUFACTURING\*\*\*\*\*

ORDER NUMBER: ORDNUM DATE: ORDATE\_\_\_\_\_

SHIP TO:  
  
 NAME: NAME\_\_\_\_\_

ADDRESS: ADDRESS\_\_\_\_\_

CITY: CITY\_\_\_\_\_STATE:\_ST

ZIP: ZIP\_\_\_\_\_ TELEPHONE: PHONE\_\_\_\_\_

QTY	PART NO	DESCRIPTION	UNIT PRICE	TOTAL PRICE
*****	*****	*****	*****	*****

Field: ORDNUM  
 Num: 8 Len: 6 Name: ORDNUM Enh: HI FType: D DType: DIG  
 Init Value:  
 \*\*\* PROCESSING SPECIFICATIONS \*\*\*  
 INIT  
 SET ORDNUM TO SORDNUM \Move order number to this field  
 SET STOTNET TO 0 \Initialize totals save field  
 SET LINE\_COUNT TO 0 \Initialize counter for order lines

Field: ORDATE  
 Num: 1 Len: 17 Name: ORDATE Enh: HI FType: D DType: MDY  
 Init Value:  
 \*\*\* PROCESSING SPECIFICATIONS \*\*\*  
 INIT  
 SET ORDATE TO \$TODAY \Assign today's date to field

Field: NAME  
 Num: 2 Len: 42 Name: NAME Enh: HI FType: R DType: CHAR  
 Init Value:  
 \*\*\* PROCESSING SPECIFICATIONS \*\*\*  
 SET SNAME TO NAME \Save customer name for ORDER\_TOTALS

Figure A-1. Forms File Listing, (cont'd)

```
FORMSPEC VERSION B.04.12          FRI, AUG 8, 1986,  2:24 PM
FORMS FILE: ORDFORM1.PUB.W          PAGE 3

FORM NAME:  ORDER_HEAD          EXECUTE THIS FORM FIRST

Field: ADDRESS
  Num: 3    Len: 42    Name: ADDRESS          Enh: HI    FType: 0  DType: CHAR
  Init Value:

Field: CITY
  Num: 4    Len: 30    Name: CITY          Enh: HI    FType: 0  DType: CHAR
  Init Value:

Field: ST
  Num: 5    Len: 2     Name: ST          Enh: HI    FType: 0  DType: CHAR
  Init Value:
*** PROCESSING SPECIFICATIONS ***
IN $STATE "ENTER 2-CHARACTER LEGITIMATE STATE CODE."

Field: ZIP
  Num: 6    Len: 10    Name: ZIP          Enh: HI    FType: 0  DType: CHAR
  Init Value:
*** PROCESSING SPECIFICATIONS ***
MATCH dddd[-dddd] "Enter ZIP code, must be either 5 digits" &
                  "or 9 digits and a hyphen."

Field: PHONE
  Num: 7    Len: 14    Name: PHONE          Enh: HI    FType: 0  DType: CHAR
  Init Value:
*** PROCESSING SPECIFICATIONS ***
MATCH [(ddd)]b*ddd-dddd "Enter phone number; any area code must be in " &
                       "parentheses; e.g., (408) 739-8808."
```

**Figure A-1. Forms File Listing, (cont'd)**

```
FORMSPEC VERSION B.04.12          FRI, AUG 8, 1986,  2:24 PM
FORMS FILE: ORDFORM1.PUB.W          PAGE 4
```

```
Form: ORDER_LINE
Repeat Option: A

Next Form Option: C
Next Form: ORDER_TOTALS
Reproduced From:
```

```
Comments: 2ND FORM-REPEATED AUTOMATICALLY, APPENDED TO ITSELF
*****
```

\*\*\*\*\*ABC MANUFACTURING\*\*\*\*\*

```
Qty__ partnum__ desc_____ uprice_ tprice
*****
```

```
Field: qty
Num: 9 Len: 5 Name: QTY Enh: HI FType: R DType: DIG
Init Value:
```

\*\*\* PROCESSING SPECIFICATIONS \*\*\*

```
INIT
SET LINE_COUNT TO LINE_COUNT + 1 \Increment line counter
IF LINE_COUNT GE 7 THEN \If more than 7 order lines
CHANGE CFORM TO NOREPEAT \Terminate repeat
```

```
FIELD
GE 1 "Enter quantity equal to or greater than 1."
```

```
Field: partnum
Num: 10 Len: 9 Name: PARTNUM Enh: HI FType: R DType: CHAR
Init Value:
```

\*\*\* PROCESSING SPECIFICATIONS \*\*\*

```
MATCH Puddd-%%? "First letter must be P, then letter A-Z, 3 digits, " &
"hyphen, 3 characters."
```

```
Field: desc
Num: 11 Len: 43 Name: DESC Enh: HI FType: O DType: CHAR
Init Value:
```

```
Field: uprice
Num: 12 Len: 7 Name: UPRICE Enh: HI FType: R DType: NUM2
Init Value:
```

\*\*\* PROCESSING SPECIFICATIONS \*\*\*

```
JUSTIFY RIGHT
STRIP LEADING "0" \Format unit price
```

```
Field: tprice
Num: 13 Len: 7 Name: TPRICE Enh: HI FType: D DType: NUM2
Init Value:
```

**Figure A-1. Forms File Listing, (cont'd)**

```

FORMSPEC VERSION B.04.12                      FRI, AUG 8, 1986, 2:24 PM
FORMS FILE: ORDFORM.PUB.W                      PAGE 5

FORM NAME: ORDER_LINE          2ND FORM-REPEATED AUTOMATIALLY, APPENDED TO ITSELF

*** PROCESSING SPECIFICATIONS ***

IF EQ $EMPTY THEN                          \In Collect mode,
  SET TPRICE TO QTY * UPRICE                \ calculate line price
  SET STOTNET TO STOTNET + TPRICE           \ and accumulate it in save field.

ELSE                                         \In Browse mode
  SET STOTNET TO STOTNET - TPRICE          \ subtract old line price
  SET TPRICE TO QTY * UPRICE              \ calculate new line price
  SET STOTNET TO STOTNET + TPRICE         \ and accumulate it in save field.

```

**Figure A-1. Forms File Listing, (cont'd)**

```

FORMSPEC VERSION B.04.12                      FRI, AUG 8, 1986, 2:24 PM
FORMS FILE: ORDFORM1.PUB.W                    PAGE 6

Form: ORDER_TOTALS
Repeat Option: N

Next Form Option: C
Next Form: $HEAD
Reproduced from:

Comments: 3RD FORM OF A 3-PART ORDER TRANSACTION
*****

Order Number: [ordnum]                      Date: [ordate_____]
Total Net Price: [totnet_____]
Discount % : [dp]                          Discount: [discnt_____]
Subtotal: [sub_____]
Tax Rate : [tr_]                            tax: [tax_____]
Shipping: [ship_____]
TOTAL: [total_____]

Customer Name: [name_____]
*****

Field: ordnum
Num: 1 Len: 6 Name: ORDNUM Enh: HI FType: R DType: DIG
Init Value:
*** PROCESSING SPECIFICATIONS ***
INIT
SET ORDNUM TO SORDNUM                      \Move order number from save fields

IF SORDNUM GT 999999 THEN                  \Terminate before 1,000,000 orders
CHANGE NFORM TO $END
ELSE SET SORDNUM TO SORDNUM + 1           \Otherwise, increment order number

Field: ordate
Num: 2 Len: 17 Name: ORDATE Enh: HI FType: D DType: MDY
Init Value:
*** PROCESSING SPECIFICATIONS ***
INIT
SET ORDATE TO $TODAY                      \Assign today's date

```

Figure A-1. Forms File Listing, (cont'd)

```
FORMSPEC VERSION B.04.12                      FRI, AUG 8, 1986, 2:24 PM
FORMS FILE: ORDFORM1.PUB.W                      PAGE 7

FORM NAME:   ORDER_TOTALS           3RD FORM OF A 3-PART ORDER TRANSACTION

Field: totnet
  Num: 3   Len: 10   Name: TOTNET           Enh: HI   FType: D   DType: NUM2
  Init Value:
*** PROCESSING SPECIFICATIONS ***
INIT
  SET TOTNET TO STOTNET                   \Set to accumulated totals (collect)

FIELD
  SET TOTNET TO STOTNET

Field: dp
  Num: 4   Len: 2   Name: DISCNT_PERCNT     Enh: HI   FType: O   DType: NUM0
  Init Value: 10

Field: discnt
  Num: 5   Len: 10   Name: DISCNT           Enh: HI   FType: D   DType: NUM2
  Init Value:
*** PROCESSING SPECIFICATIONS ***
IF TOTNET GE 1000 THEN                       \If net price exceeds $999.00
  SET DISCNT TO DISCNT_PERCNT % TOTNET       \Calculate discount
ELSE SET DISCNT TO 0                         \Otherwise, set discount to zero

Field: sub
  Num: 6   Len: 10   Name: SUB              Enh: HI   FType: D   DType: NUM2
  Init Value:
*** PROCESSING SPECIFICATIONS ***
SET SUB TO TOTNET - DISCNT                   \Calculate subtotal

Field: tr
  Num: 7   Len: 4   Name: TAX_RATE          Enh: HI   FType: R   DType: NUM2
  Init Value: 6.25

Field: tax
  Num: 8   Len: 10   Name: TAX              Enh: HI   FType: D   DType: NUM2
*** PROCESSING SPECIFICATIONS ***
SET TAX TO TAX_RATE % SUB                   \Calculate tax

Field: ship
  Num: 9   Len: 10   Name: SHIP            Enh: HI   FType: O   DType: NUM2
  Init Value: .00
*** PROCESSING SPECIFICATIONS ***
JUSTIFY RIGHT
STRIP LEADING "0"                           \Format Shipping cost
```

Figure A-1. Forms File Listing, (cont'd)

FORMSPEC VERSION B.04.12 FRI, AUG 8, 1986, 2:24 PM  
FORMS FILE: ORDFORM1.PUB.W PAGE 8

FORM NAME: ORDER\_TOTALS 3RD FORM OF A 3-PART ORDER TRANSACTION

Field: total  
Num: 10 Len: 10 Name: TOTAL Enh: HI FType: D DType: NUM2  
Init Value:  
\*\*\* PROCESSING SPECIFICATIONS \*\*\*  
SET TOTAL TO SUB + TAX + SHIP \Calculate total for order

Field: Name  
Num: 11 Len: 47 Name: NAME Enh: HI FType: D DType: CHAR  
Init Value:  
\*\*\* PROCESSING SPECIFICATIONS \*\*\*  
SET NAME TO SNAME \Move customer name from save field



## Appendix B Glossary of Terms Used in this Guide

appended form	A form that is displayed on the terminal screen immediately below another form.
arithmetic expression	Any constants, field names, or save field names combined by arithmetic operators and resulting in a numeric value that can be used as an operand in a FORMSPEC processing statement.
arithmetic operator	Any of the symbols (+, -, *, / or %) used in an arithmetic expression to indicate type of operation combining constants, field names, or save field names.
batch file	An MPE disc file used to collect data entered through ENTRY.
batch record	A record in a batch file; one record is written each time ENTER is pressed to enter data in a form.
block mode	Terminal mode in which ENTER (rather than RETURN) is pressed to transfer data; allows multiple lines to be typed before being transferred.
BROWSE f7	ENTRY function key used to go to Browse mode. Physically, this is the same as COLLECT.
Browse mode	ENTRY mode in which user can review and/or modify records previously written to the batch file.
Browse/Modify mode	See Browse mode.
CHAR	Data type indicating a character type field.
character type field	A field in a FORMSPEC form into which any characters may be entered.
COLLECT f7	ENTRY function key used to return to Collect mode from Browse mode. Physically, this is the same as BROWSE.
Collect mode	ENTRY mode in which user enters data through FORMSPEC forms and writes the data to batch file records.
comment	A descriptive comment entered by the FORMSPEC designer on the Form Menu to describe the form, or in the Processing Specifications area of the Field Menu to

describe a particular statement; a comment is listed with the forms file, but does not appear at forms file execution.

compile a forms file      Make the forms file ready for use; each time a forms file is changed it must be re-compiled so that the changes are available to an application or to ENTRY.

conditional processing      Type of FORMSPEC processing statement that is executed only when a condition is true; conditional processing is always introduced by an IF statement.

constant                      A literal value used in FORMSPEC processing statements; character type constants are surrounded by quotes, data type constants by exclamation points, numeric constants do not use delimiters.

custom error message      A message associated with a particular field edit to be displayed in the window line of a form when a field fails the edit; this message is issued only if the field is in error and is the first field in screen order.

data                              Values entered through an application or ENTRY into forms designed by FORMSPEC. ENTRY then writes the data to the batch file.

data collection              The process of entering data through ENTRY and writing that data to batch file.

data formatting              The reorganization of entered data according to automatic formatting rules or as a result of formatting statements.

data type                      Determines type of data that can be accepted in a field; may be character type, a numeric type, or a data type.

data type field              A field in a FORMSPEC form into which only a date may be entered; specific date types are DMY, MDY, and YMD.

DELETE REC    f2              ENTRY function key used to delete the currently displayed record from the batch file.

delimiters                      Symbols used to mark the start and end of each field on the screen for a FORMSPEC form; standard delimiters are opening and closing brackets ([ and ]); non-printing delimiters may be entered by using ESCAPE with brackets.

DIG                              Data type used to indicate a numeric type field into which only digits (whole numbers) can be entered.

display-only field              A field in a FORMSPEC form that does not accept data entered by the user; data is moved to such fields as a

result of FORMSPEC processing or initialization.

DMY Data type used to indicate a data type field into which the entered date must be in the order: Day Month Year.

edit The process of checking entered data according to field type, data type or an edit statement associated with the field.

edit statement One of a set of FORMSPEC processing statements that tests entered data.

ENTER Key used to transfer data to a program executing in block mode; used in ENTRY to transfer data to batch file, in FORMSPEC to transfer the source version of the forms file.

ENTRY Program that displays FORMSPEC forms on terminal screen and accepts data entered by the user into these forms, then writes data to batch file.

EXIT f8 FORMSPEC and ENTRY function key used to terminate execution of either of these programs and return to MPE control.

expression Combination of operators and operands that results in a single value; in FORMSPEC processing, expressions are always arithmetic.

fast forms file File to which forms file is compiled when it is ready for use in a production environment; because record size is compacted, file executes more quickly than regular forms file; however, fast forms file cannot be modified.

field Area designated on FORMSPEC form to contain data subsequently entered through an application or ENTRY.

FIELD Header in FORMSPEC processing specifications that introduces statements to be executed in field edit phase.

field edit See edit.

field edit phase Phase of FORMSPEC processing in which data entered in each field is edited; normal mode of execution.

field length Length in number of characters of a FORMSPEC field; field length is determined by placement of delimiters during form layout.

Field Menu FORMSPEC menu in which designer specifies field attributes of each field and/or enters processing

statements that control how data entered in field is treated.

field name	Name assigned to each field in a FORMSPEC form. The name may be up to 15 uppercase characters. The first character must be alphabetic; the other characters can be alphabetic, digits, or the underline (_). Each field name must be unique to the form of which it is a member.
field tag	Name entered between field delimiters to identify field during FORMSPEC form layout; tag may be upper or lower case letters, digits 0-9, or underline, it is subsequently shifted to all uppercase to form the default field name.
FIELD TOGGLE f3	FORMSPEC function key used to go to the Processing Specification area of a Field Menu in order to enter processing statements.
field type	Determines how data entered into field will be processed--whether data is optional or required, display-only, or to be processed even if blank.
file	A collection of related records treated as a unit; VPLUS/V files include the FORMSPEC forms file and the ENTRY batch file.
finish phase	Final phase of processing performed on a form after all data entered into the form has been accepted.
FINISH	Header that introduces statements to be executed during finish phase of FORMSPEC processing.
FIRST REC f1	ENTRY function key used in Browse mode to display first record in batch file.
form	An ordered arrangement of text and fields for data entry, saved in a forms file for subsequent display on a terminal screen; form design is controlled by FORMSPEC, entry of data through the form by an application or by ENTRY.
form layout	Physical arrangement of text and fields in a FORMSPEC form showing how the form will appear when displayed at the terminal.
Form Menu	FORMSPEC menu through which forms designer specifies characteristics of each form, such as form name and form sequence.
form name	Unique name assigned to each form in FORMSPEC forms file; name may be 1-15 uppercase characters, A-Z,0-9, or

\_, must start with letter A-Z.

form sequence      Sequence in which FORMSPEC forms are displayed, also, and whether a form repeats and/or is appended to previous form.

forms file          A MPE file containing definitions of a set of related forms that are displayed by an application or by ENTRY.

Forms File Menu    FORMSPEC menu through which forms designer requests that  
that  
a new forms file be created to contain a new set of forms, or that an existing forms file be accessed so that it can be modified.

FORMSPEC          Program through which forms designer designs a set of related forms and saves them as a forms file, or modifies the forms in an existing forms file.

frozen form        Form that remains (is frozen) on the screen when the next form in form sequence is displayed below it.

function keys      Set of keys on HP terminals that are programmed to perform particular functions; the functions of these keys differ for FORMSPEC, ENTRY Browse mode, and ENTRY Collect mode.

globals            Characteristics of a FORMSPEC forms file that apply to all forms in the file, rather than being local to a particular form.

Globals Menu       FORMSPEC menu on which the forms designer specifies non-default global characteristics of the forms file.

head form          First form displayed when forms file is executed; it need not be first form in file.

HEAD FORM f1      ENTRY function key used to display head form on terminal screen.

IMPn               Numeric data type indicating field with implied decimal point; *n* decimal positions; allows numeric data with sign, commas, decimal point.

index-retrieve    List of operands one of which is selected according to operand  
operand            the value of an index.

INIT               Header introducing statements to be executed in the initialization phase of FORMSPEC processing.

initial value      Value assigned to a field or save field; appears in field when the form containing the field is first

displayed.

initialization phase	Initial phase of FORMSPEC processing performed before form is displayed on terminal screen.
list device	An output device, such as a line printer, to which listings of program or data files are directed.
listing	Printed output showing the external characteristics of a file or form; forms file listings are generated by FORMSPEC, form or batch file listings by ENTRY.
literal value	Exact value, rather than an expression or field name, that indicates a type of value; see constant.
MAIN f7	FORMSPEC function key used to request the Main Menu.
Main Menu	FORMSPEC menu on which forms designer requests a form design function, such as adding a form, listing forms file, compiling forms file.
MDY	Data type used to indicate date type field in which entered date must be in order: Month Day Year.
menu	Form issued by FORMSPEC on which form designer requests particular functions or enters form characteristics.
message	Descriptive phrase issued in window line of a form to provide documentation, warnings, or error diagnosis.
MPE	The Multiprogramming Executive operating system of the HP 3000 computer system.
NEXT f6	FORMSPEC function key used to display next menu in the sequence of FORMSPEC menus.
NEXT FORM f2	FORMSPEC function key used to display next Form Menu.
NEXT FORM f6	ENTRY function key used to stop a repeating form and display next different form.
NEXT REC f6	ENTRY function key used to display next batch record.
NUM	Numeric data type that causes data entered in field to be formatted as floating point number; allows numeric data with sign, commas, decimal point.
NUMn	Numeric data type indicating field with <i>n</i> decimal positions, actual decimal point; allows numeric data with sign, commas, decimal point.
operating	System that controls user interface with HP 3000

system computer system; specifically controls program operation and files.

operand Element used in FORMSPEC processing statements; depending on statement may be constant, field name, save field name, arithmetic expression, or index retrieve operand.

optional field Field in FORMSPEC form that allows data to be omitted.

phase Specification that determines when FORMSPEC processing is to be performed: before data entry, during data entry, or following data entry.

PREV f5 FORMSPEC function key used to display previous menu in sequence of FORMSPEC menus.

PREV FORM f1 FORMSPEC function key used to display previous Form Menu.

PREV REC f5 ENTRY function key used to display previous batch record.

PRINT f3 ENTRY function key used to list form or batch record on line printer.

process field Field in FORMSPEC form that is processed even if no data is entered (field is blank).

processing specifications area Area in FORMSPEC Field Menu in which processing statements can be entered.

processing statement Statement that describes processing to be performed on data entered in a field; statement is entered in processing specification area of Field Menu.

protected field A field in a FORMSPEC form that is protected from data entry.

REFRESH f4 FORMSPEC function key used to clear FORMSPEC menu of any specifications entered by designer and redisplay cleared menu.

REFRESH f4 ENTRY function key used to clear displayed form of entered data.

repeating form A form that is repeated on terminal screen until the user presses the NEXT FORM key.

required field A field in a FORMSPEC form into which data must be

entered.

RESUME f7 FORMSPEC function key that returns to the menu displayed before designer pressed MAIN to go to Main Menu. Physically, the same as MAIN.

RETURN Key used to enter data or a command when terminal is not in block mode.

save field A field in a FORMSPEC forms file that is not part of any form, and can be used to hold (save) data used by any field or form in the file.

screen order Order of fields in a screen in which first field is at top left of screen and last field at bottom right.

source file Version of FORMSPEC forms file that can be modified as opposed to the compiled version of the same file that can be executed.

statement Combination of a key word and parameters that describes processing to be performed on field in FORMSPEC form; see processing statement.

unprotected field Field in a FORMSPEC form into which the user can enter data.

window Line in FORMSPEC menu or form displayed by an application or by ENTRY in which message is displayed.

YMD Data type used to indicate date type field in which entered date must be in order: Year Month Day.

}

---

[ERROR: ] [ERROR: ]

---