IBM

SR30-0069-1

File No. S32-32

GENERAL
SYSTEMS
DIVISION
EDUCATION

IBM System/32
Data File Utility (DFU)
Facilities and Operations
Student Text

SR30-0069-1

File No. S32-32

GENERAL
SYSTEMS
DIVISION
EDUCATION

IBM System/32
Data File Utility (DFU)
Facilities and Operations
Student Text

Second Edition (February 1978)

Requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality. Address comments concerning this publication to: IBM Corporation, General Systems Division, Education Development, Dept. 71G,
P. O. Box 2150, Atlanta, Georgia 30301

Comments and suggestions become the property of IBM.

© Copyright International Business Machines Corporation 1975

TABLE OF CONTENTS

THIS PAGE LEFT INTENTIONALLY BLANK

INTRODUCTION

SOME THOUGHTS ABOUT SELF-STUDY

What is self-study?

Self-study is a method of presenting information which allows a student to study independently and proceed at his own pace. The subject matter is carefully organized into a logical sequence and presented in discrete steps. The student participates actively at every step.

By what methods can self-study courses be presented?

Textbooks are the most commonly used medium for presenting self-study courses. Other methods include audio tapes, video tapes, motion pictures and even live teachers. The choice of medium depends on the activities the student will engage in for the most efficient learning. This course will use one other medium and that is the system itself.

What principles of self-study should a student know?

These are some fundamental principles that the student should know in order to use a self-study course effectively.

1. Logical Sequence. The order of topics in the textbook assures that information is presented to the student only when he is ready to learn it.

2. Discrete Steps. The material is presented in easily learned segments. The student concentrates on one topic at a time and is never forced to absorb a mass of facts all at once.

3. Active Responding. People learn by doing, not by being "lectured at" and not by just reading. Therefore, each topic requires the student to respond by doing something with each fact.

4. Immediate Confirmation. In this self-study course the student responds by operating the system. The text will direct the student through an operation, give him a chance to do it, then tell him what should have happened.

5. Self-pacing. Not all people learn at the same rate. Self-study lets the student pick the pace that suits him.

Do self-study courses bring the student to an end point or is further instruction required?

Each course is a complete instructional package. However, no course tries to exhaust its subject, and the student is required to learn more on the job by actually applying what he has learned through self-study. In some cases the student is directed to attend a follow-up classroom session.

What materials are provided for self-study courses?

The materials vary somewhat. The main types of materials are:

1. Text. The self-study courses are built around a student text which for most courses is a single book. In general, the textbooks are meant to be reused.

2. Illustrations. Many courses have a separate book or books of illustrations, also reusable; however, some have illustrations printed in the text, just like conventional textbooks.

3. Student Materials. Some courses provide materials for the student to keep when he completes a course. Examples of student materials found in different courses are:

    a. Notebook. A special booklet in which the student is directed to write specified things during his study of the text.

    b. Problem Book. A workbook containing problems, exercises, and quizzes.

    c. Reference Handbook. A reference book designed to be used in conjunction with a programmed text.

4. Examination. Some courses provide final examinations; some also provide intermediate examinations.

What does a self-study text look like?

There is no standard format for self-study texts. The format will depend on the medium used in the course.

COURSE DESCRIPTION

SYSTEM/32 DATA FILE UTILITY (DFU) FACILITIES AND OPERATIONS

INTRODUCTION

This self-study course is designed to teach you how to operate the System/32 using DFU. It is a self-study course where you will learn by reading material and by doing exercises on the system.

Prerequisite:

The student is expected to be able to operate the System/32. Attendance at the Data Processing Concepts Course (Y2050) or equivalent experience is recommended.

What material is used in this course?

The self-study course consists of:

Self-Study Text	SR30-0069

In addition to the above material, the student must have access to the following system manuals:

IBM System/32 Utilities Program Product Reference Manual - Data File Utility	SC21-7600
IBM System/32 Messages Guide - Utilities Program Product	SC21-7618
IBM System/32 Utilities Program Product Reference Manual - Source Entry Utility	SC21-7605
IBM System/32 Operator's Guide	GC21-7593

What is the main strategy of the course?

The course is based on the idea that experience is the best teacher. Therefore, this hands-on self-study course combines the ideas of hands-on experience and the learner-paced format of self-study.

Who should study this course?

System operators, programmers, implementers and others who have a need to learn and apply DFU facilities.

What is the time period within which the student is expected to complete the course?

Study time will require approximately 6 hours for most students.

Self-study courses are not intended to be studied eight hours a day, and students should be aware of this before beginning the course. Gaps are likely to occur in the student's schedule, but interruptions of a day or more should not occur in the study program. Each topic has an estimated required study time. Break points should be taken between topics.

How self-study should be studied.

1. Self-study requires careful concentration. You should not be disturbed during your study.

2. Do not try to memorize as you read and do not take notes as you proceed through the text.

3. Try to iron out any difficulties for yourself. If you have trouble, you will learn more if you struggle with it than if you call for help. It is permissible to reread previous sections, or to read ahead, in order to figure out the answer to a problem.

4. No marks of any kind should be made in the student text if the course will be given again to another student.

Course Content

This course consists of seven exercises which will use all the facilities of the Data File Utilities for the System/32, hereafter referred to as DFU.

The seven exercises will include loading two data files, updating a data file, inquiring into a data file, and three exercises which will list one or both of the two data files previously loaded. The two data files which we will use in this course consist of:

1. an inventory master file, and

2. a transaction file which would update the inventory master file.

In addition, a preliminary step will be taken to prepare the System/32 for the running of the DFU exercises. This will involve using another IBM utility named the Source Entry Utility (SEU) to load two RPG II modules, representing the two data files.

STUDY SCHEDULE

(NOTE: The estimated study times are average times. A student may take more or less time for any one topic.)

STUDY ITEMS	ESTIMATED TIMES
INTRODUCTION TO COURSE	15 mins
SYSTEM PREPARATION	50 mins
EXERCISE 1	60 mins
EXERCISE 2	30 mins
EXERCISE 3	30 mins
EXERCISE 4	20 mins
EXERCISE 5	20 mins
EXERCISE 6	30 mins
EXERCISE 7	45 mins
	5 hrs

The timings on exercises tend to shorten as the course proceeds. This is largely due to the student becoming more familiar with DFU rather than to the relative complexity of the exercises.

Including normal break periods between exercises, the course duration should be approximately six hours exclusive of lunch.

During the course you will be creating two files of data. In addition, you will store other information in the computer for reference.

WARNING: The data files and the other information stored for use during this course may cause a conflict with similar items used by your company (the names might be the same). If your company uses data files named either INVENT or TRANS on the IBM System/32, use other names of your choice throughout this course. Likewise, if your company uses library member names like those that follow, use different ones during the course.

SOURCE MEMBERS	LOAD MEMBERS	
QINVEN	QEX1	QEX5
QTRANS	QEX2	QEX6
QEX7S	QEX3	QEX7
	QEX4	QEX8

THIS PAGE LEFT INTENTIONALLY BLANK

SYSTEM PREPARATION
(SOURCE MEMBER LOADING)

Before we begin the seven exercises, we must load two RPG II source members each of which will represent one of the two data files. These source members will describe each of the data files and the records contained therein. Later, when we run our DFU exercises, we will be referring to these source members in order to run our programs.

At this time familiarize yourself with Chapter 8 of the DFU Reference Manual (SC21-7600). It is titled 'RPG II Source Member' and gives a brief overview of the type of RPG II entries we will be making.

In order to enter these source members, we must use the System/32 Source Entry Utility, commonly referred to as SEU. We will guide you through the entry of these source members using SEU. In addition, you should have available the IBM System/32 Utilities Program Product Reference Manual - Source Entry Utility (SC21-7605) for reference as we proceed entering the source members.

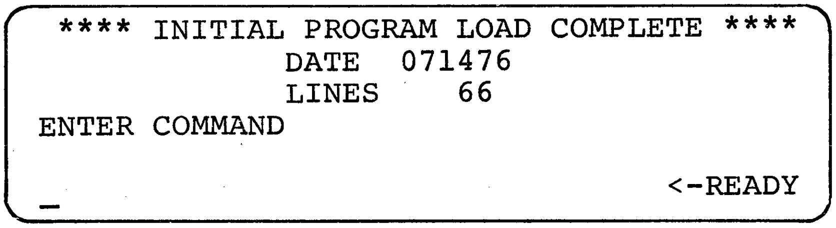
***

NOTE: It is not an objective of this course to teach SEU. However, if you are not already experienced with it, the practice obtained by loading these two members plus a familiarization with the SEU Reference Manual should enable you to load your own source members after you have finished the course.

Often somebody else in your department will have experience in RPG II and SEU and should be able to help you, if necessary. If your company is using an IBM supplied package, these source members may already be on your system.

***

If you have not already powered on the system and IPL'ed, please do so now. Upon completion of the IPL, your screen should have a display similar to that shown in Figure 1.1.



```
 **** INITIAL PROGRAM LOAD COMPLETE ****
           DATE  071476
           LINES    66
ENTER COMMAND

                                 <-READY
_
```

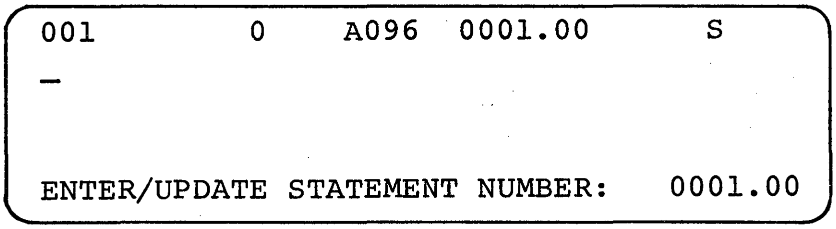
Figure 1.1

At this point, use the DATE command to key in today's date, if appropriate. For example, DATE 071576 indicates a date of 7/15/76.

Now it is time to enter the first source member for our inventory master file. To do this, type in the command:

```
SEU QINVEN,R
```

Then, press the Enter Key.

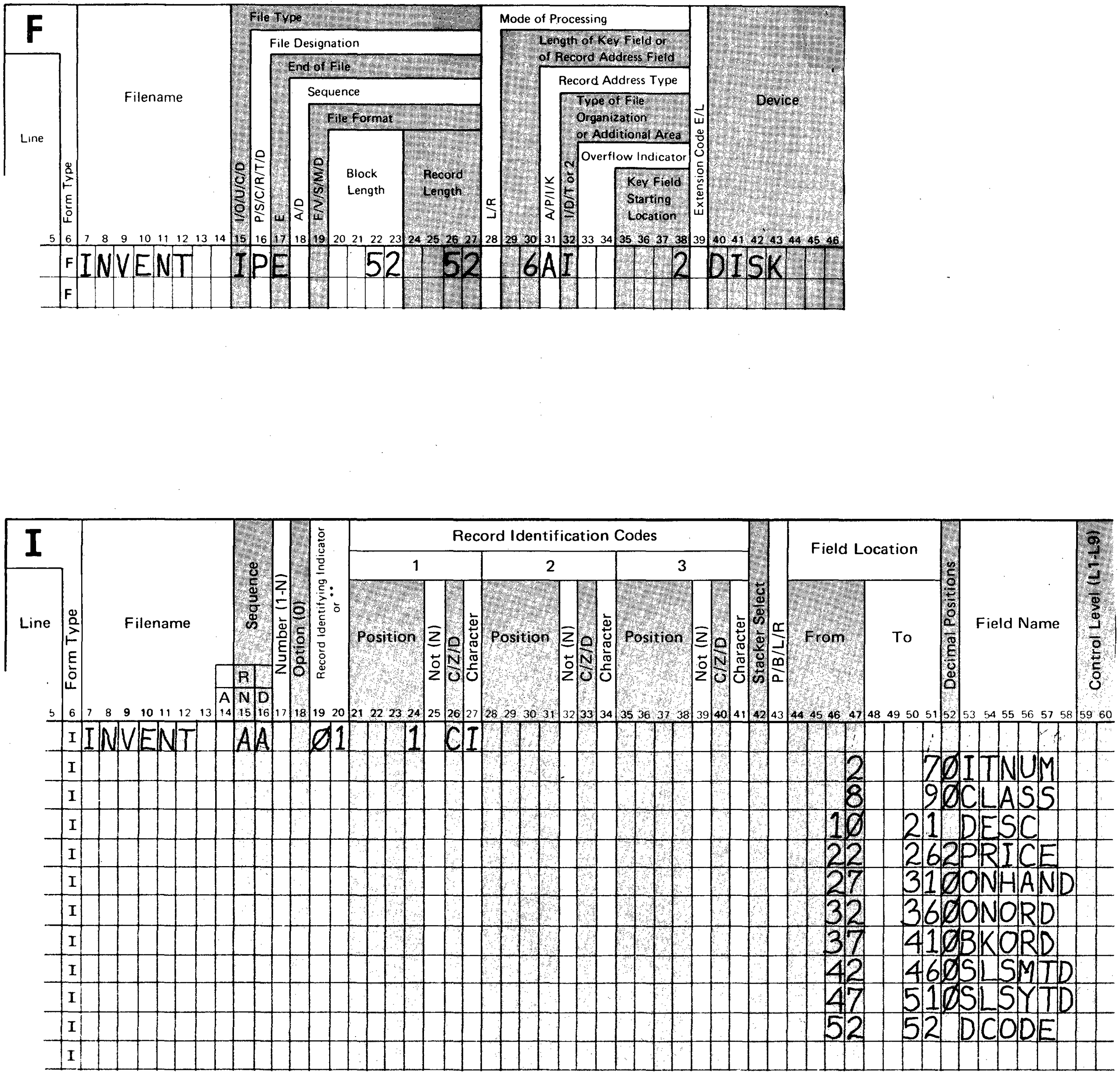


```
001          0     A096  0001.00       S
_


ENTER/UPDATE STATEMENT NUMBER:    0001.00
```

Figure 1.2

When the Keyboard Ready Light comes back on in about 10 or 15 seconds, your screen will have a display identical to that shown in Figure 1.2.

The command just entered calls for the procedure SEU to be retrieved from disk and to be executed. The parameters of the SEU command include QINVEN, which is the name that we will give to the source member after we have finished entering the data. The R appearing after the comma simply indicates that QINVEN will be an RPG II source member. Now, look at Figure 1.3.



F

Line	Form Type	Filename	File Type I/O/U/C/D	File Designation P/S/C/R/T/D	End of File E	Sequence A/D	File Format F/V/S/M/D	Block Length	Record Length	Mode of Processing L/R	Length of Key Field or of Record Address Field	Record Address Type A/P/I/K	Type of File Organization or Additional Area I/D/T or 2	Overflow Indicator	Key Field Starting Location	Extension Code E/L	Device
	F	INVENT	I	P	E			52	52		6	A	I		2		DISK
	F																

I

Line	Form Type	Filename	Sequence	Number (1-N)	Option (O)	Record Identifying Indicator or **	Record Identification Codes 1 Position	Not (N)	C/Z/D	Character	2 Position	Not (N)	C/Z/D	Character	3 Position	Not (N)	C/Z/D	Character	Stacker Select	P/B/L/R	Field Location From	To	Decimal Positions	Field Name	Control Level (L1-L9)
	I	INVENT	AA			Ø1	1		C	I															
	I																				2	7	Ø	ITNUM	
	I																				8	9	Ø	CLASS	
	I																				1Ø	21		DESC	
	I																				22	26	2	PRICE	
	I																				27	31	Ø	ONHAND	
	I																				32	36	Ø	ONORD	
	I																				37	41	Ø	BKORD	
	I																				42	46	Ø	SLSMTD	
	I																				47	51	Ø	SLSYTD	
	I																				52	52		DCODE	
	I																								

Figure 1.3

This shows the data that we will input into the source member QINVEN. It is an RPG II representation of the structure of the inventory master file and the records therein, as shown in Figure 1.4.

INVENTORY MASTER FILE

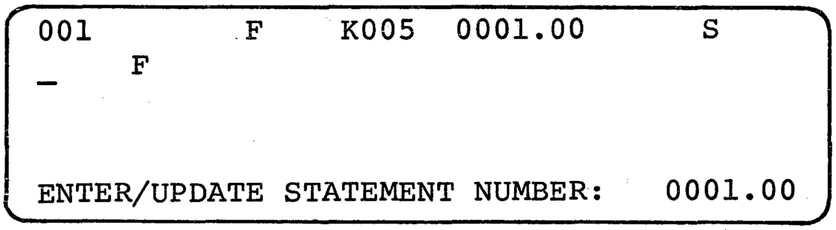
RPG II FIELD NAME	POSITION WITHIN RECORD	DECIMAL POSITIONS	FIELD DESCRIPTION
'I'	1	Alpha	Record Identification Code
ITNUM	2-7	0	Item Number
CLASS	8-9	0	Item Class
DESC	10-21	Alpha	Item Description
PRICE	22-26	2	Item Price
ONHAND	27-31	0	Units On Hand
ONORD	32-36	0	Units On Order
BKORD	37-41	0	Units Back Ordered
SLSMTD	42-46	0	Unit Sales Month-to-date
SLSYTD	47-51	0	Unit Sales Year-to-date
DCODE	52	Alpha	Delete Code

Figure 1.4

Notice in Figure 1.3 that the first line has a pre-printed F in Column 6, indicating that that line is a File Description specification, a certain type of RPG II entry. All other lines have an I in Column 6, indicating that we are describing the input with those lines (Input specifications).

In order to enter the first line, the File Description line (F in Column 6), we must do the following: press Command Key 3 (which indicates that we wish to enter a new type of line), key the letter F, and depress the Enter button. This sets the screen up to receive RPG II specifications in the format of a File Description line which we are now about to enter.

Look at Figure 1.5.



```
001          F    K005  0001.00        S
_      F



ENTER/UPDATE STATEMENT NUMBER:    0001.00
```

Figure 1.5

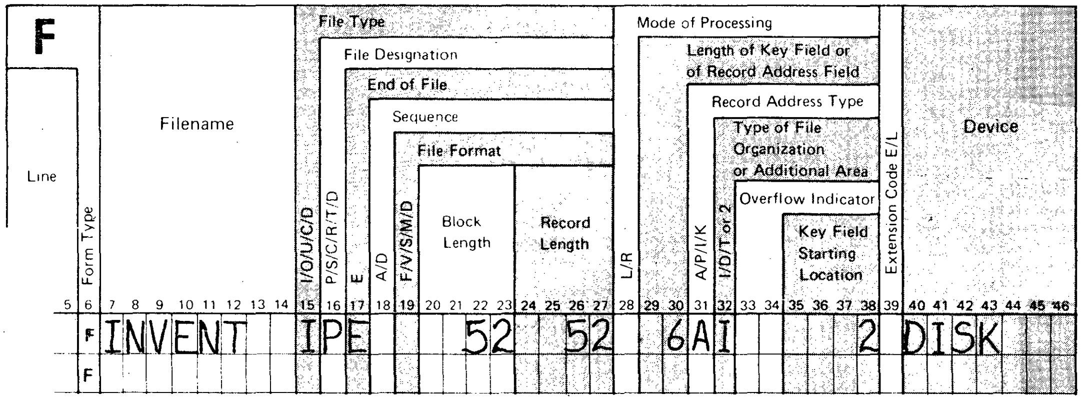
First, notice the cursor is now positioned in position 1 of the second line. The F that appears to the right of the cursor is in position 6. As the F is a standard entry for every File Description line, it is already inserted correctly by SEU.

Now press Command Key 6.

This will cause each line of entry to be printed on the printer for visual checking purposes. Notice the P (for printing) which appears in the upper right hand corner of the screen as a result of pressing Command Key 6.

Notice also three of the fields on the display screen. First, in the upper left hand corner, is the number 001. This number reflects the position within the record where the cursor is at this time and where any keying that we are about to do will appear on the screen. Further to the right, K005 refers to the fact that SEU is now set up to receive a five-position numeric field. Immediately to the right of that, 0001.00 is the number of the statement that we are presently entering (Statement 1). These figures will change as we proceed and should be referenced as we make our entries.

Now, as the cursor is in position 1 and a five-position field is set up for entry, look at Figure 1.6 (a repetition of the first line from Figure 1.3 included here for reference purposes).


F
Line
Form Type
Filename
File Type
File Designation
End of File
Sequence
File Format
Block Length
Record Length
I/O/U/C/D
P/S/C/R/T/D
E
A/D
F/V/S/M/D
L/R
Mode of Processing
Length of Key Field or of Record Address Field
Record Address Type
Type of File Organization or Additional Area
Overflow Indicator
Key Field Starting Location
A/P/I/K
I/D/T or 2
Extension Code E/L
Device
5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
F

F INVENT IPE 52 52 6AI 2 DISK

Figure 1.6

We do not wish to make an entry in positions 1 to 5. Therefore, press the Enter button. The cursor moves to position 7. Notice that now two of the three fields on the screen referred to earlier have been updated to indicate:

1. The cursor is at position 7 (007).

2. SEU is ready to accept an eight-position alphameric field (A008).

3. We are still at statement 1 (0001.00).

At this point, key in INVENT which is the name we will give to our file, the inventory master file, for which we are now about to enter the source member. After INVENT is keyed press the Enter button. The cursor has moved to position 15 where it is waiting for a one-position alphabetic field. We can see at this point that an I should be entered. Do this now and then press the Enter key.

***

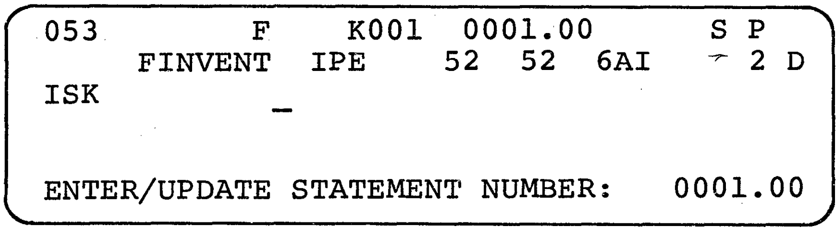
NOTE: As you are keying your entries, there will probably be times when the screen will flash indicating that SEU has detected an error in the entry just made. When this happens, simply press the Error Reset button to stop the flashing, press the Field Backspace key to reposition the cursor, if necessary, and re-key the correct entry.

***

Continue entering the rest of the line as follows (numeric fields will automatically be right-adjusted upon pressing the Enter button):

CURSOR POSITION	TYPE	PRESS
16	P	Enter
17	E	Enter
18	No Entry	Enter
19	No Entry	Enter
20	52	Enter
24	52	Enter
28	No Entry	Enter
29	6	Enter
31	A	Enter
32	I	Enter
33	No Entry	Enter
35	2	Enter
39	No Entry	Enter
40	DISK	Enter

At this point, your screen should now resemble Figure 1.7.



```
053         F     K001  0001.00        S P
     FINVENT  IPE     52   52  6AI       2 D
ISK
           _

ENTER/UPDATE STATEMENT NUMBER:    0001.00
```

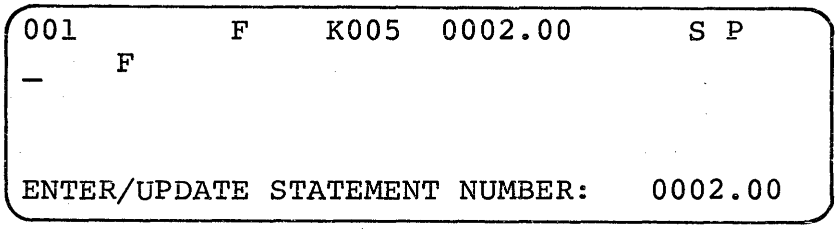
Figure 1.7

If there are any corrections to be made, press the Field Backspace key until the cursor returns to the field you wish to change. Then, make the change and press the Enter button to record it. Having made any changes necessary, press the Record Advance key. This indicates to the System/32 that we have finished entering this line. The line is now printed on our printer.

***

NOTE: There will be an opportunity later on to identify and correct any errors not detected before the Record Advance key is pressed after each line.

***

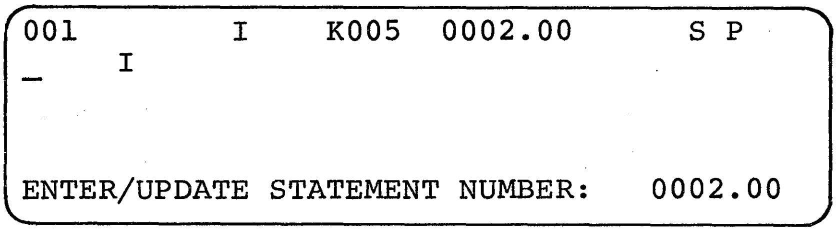


```
001         F     K005  0002.00        S P
     F
_


ENTER/UPDATE STATEMENT NUMBER:    0002.00
```

Figure 1.8

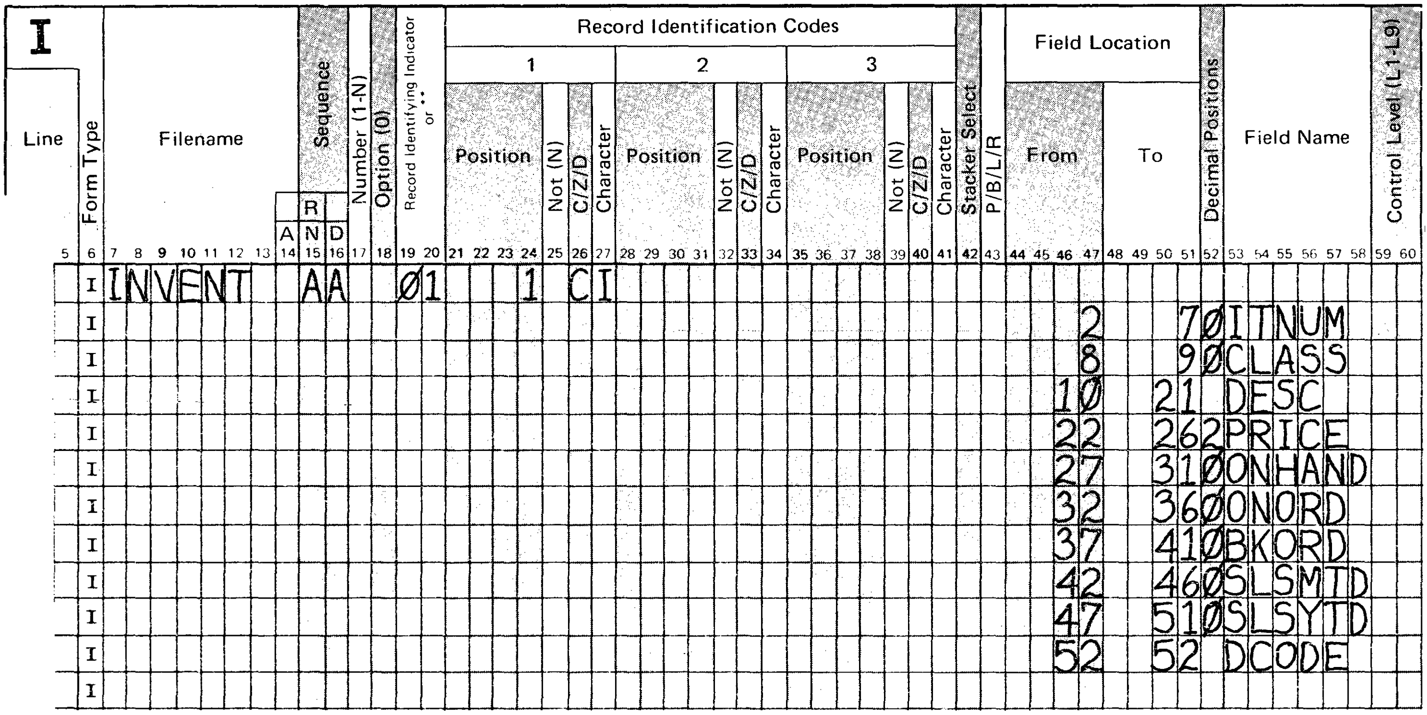
Your screen should now resemble Figure 1.8. Notice that the statement number field has been updated to 0002.00 indicating that SEU is ready to receive the next line. In addition, notice that the "F" is in position 6 indicating SEU is expecting a File Description line. However, as we have already entered the only File Description line for our entry QINVEN, we are now ready to move on to our input lines (those identified by an "I" in Column 6).

To tell the System/32 that we are now about to enter "I" lines, press Command Key 3 as before, but key the letter "I" and then press the Enter button. Your screen will then show the display illustrated in Figure 1.9.



```
001          I    K005  0002.00       S P
_     I



ENTER/UPDATE STATEMENT NUMBER:    0002.00
```

Figure 1.9



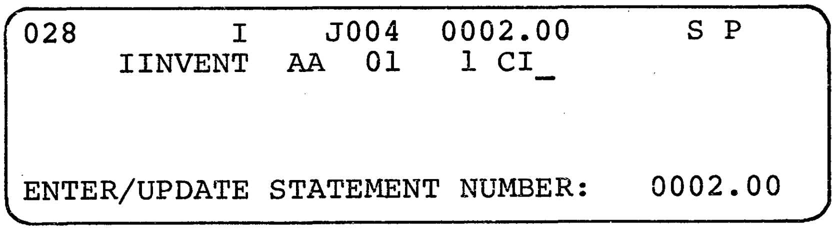
I

Line	Form Type	Filename	Sequence	Number (1-N)	Option (O)	Record Identifying Indicator or **	Record Identification Codes												Stacker Select	P/B/L/R	Field Location		Decimal Positions	Field Name	Control Level (L1-L9)
							1				2				3						From	To			
							Position	Not (N)	C/Z/D	Character	Position	Not (N)	C/Z/D	Character	Position	Not (N)	C/Z/D	Character							
	I	INVENT	AA			Ø1	1		C	I															
	I																				2	7	Ø	ITNUM	
	I																				8	9	Ø	CLASS	
	I																				1Ø	21		DESC	
	I																				22	26	2	PRICE	
	I																				27	31	Ø	ONHAND	
	I																				32	36	Ø	ONORD	
	I																				37	41	Ø	BKORD	
	I																				42	46	Ø	SLSMTD	
	I																				47	51	Ø	SLSYTD	
	I																				52	52		DCODE	
	I																								

Figure 1.10

At this point, we will key in the first line of our Input specifications (as shown in Figure 1.10). Press Enter to skip to position 7, key INVENT and press Enter. Notice that now at position 15, SEU is ready to receive a four-position alphameric field from 15 to 18. As our entry is AA,blank,blank, simply key in AA and press Enter. As this field is described as an alphameric field to SEU, there is no right adjustment.

Now key 01 and press Enter. Then 1, Enter and now the three position entry 'space, C,I', and press Enter. Again, as this is the last entry for this line, take a look at the screen to make sure that it has the entries as we intended to key them. It should be identical to Figure 1.11.

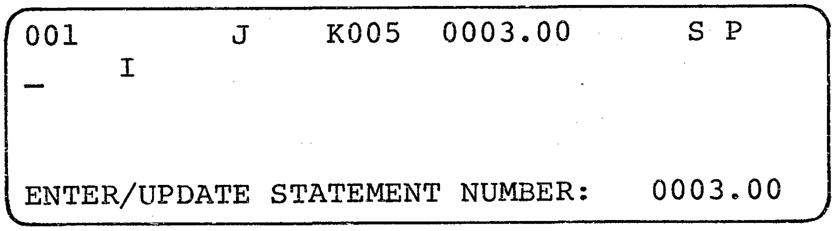


```
028          I    J004  0002.00       S P
     IINVENT  AA   01    1 CI_



ENTER/UPDATE STATEMENT NUMBER:    0002.00
```

Figure 1.11

Press the Record Advance Key after any changes have been made. Again, our statement has been printed on the printer and has been stored. Now we are to enter our next line. This line and the remaining Input lines which follow all describe fields within a data record in the file INVENT rather than the entire record.

To tell SEU that we are now about to enter these fields, press Command Key 3, the letter "J" and the Enter button. The "J" further defines our specification as a field input line as opposed to a record input line. Your screen will now show a display identical to Figure 1.12.



```
001          J    K005  0003.00       S P
_    I



ENTER/UPDATE STATEMENT NUMBER:    0003.00
```

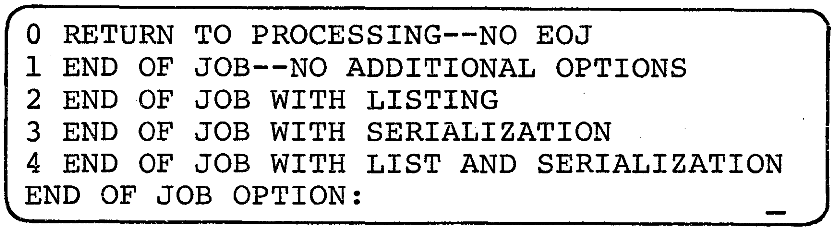
Figure 1.12

Notice that while the "I" still appears in position 6, the letter "J" now appears in the heading line instead of the "I" to indicate that we are entering data with the "J" format, that is, Field Input specifications. As our first entry for the next line begins in the field in positions 44 through 47 (as shown in Figure 1.10), press Enter twice. This moves the cursor to position 44. Key a 2 and press Enter. Again, notice the numeric field is right-adjusted. Then key a 7, Enter, a 0, Enter, ITNUM, and Enter. As this is the last field to be entered for this line, press Record Advance, having checked the screen first, to make sure that your entries are correct.

Do not worry at this time if you detected an error after you have pressed Record Advance. There will be an opportunity later to make any changes which are then required.

Now the screen is set to receive our fourth statement. Key the fourth and then the rest of the Input specifications as shown in Figure 1.10.

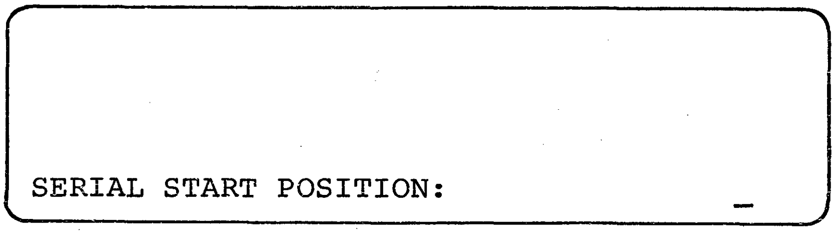
When you have finished entering the last line, press Command Key 7. Command Key 7 is used to indicate the end of job to SEU. A display as shown in Figure 1.13 appears on the screen.



```
0 RETURN TO PROCESSING--NO EOJ
1 END OF JOB--NO ADDITIONAL OPTIONS
2 END OF JOB WITH LISTING
3 END OF JOB WITH SERIALIZATION
4 END OF JOB WITH LIST AND SERIALIZATION
END OF JOB OPTION:                        _
```

Figure 1.13

On the screen are displayed a number of options we can take when we come to the end of job. For our purposes, take Option 4 which will print a list of our entries and will serialize them (that is, assign sequential serial numbers to them). Do this by keying a 4 and pressing the Enter button. Since we have chosen an option that includes serialization, the screen will now ask you for a serial start position as shown in Figure 1.14.



```
SERIAL START POSITION:                    _
```

Figure 1.14

At this point, do not key a number, but simply press the Enter button as we wish to have our serial numbers in positions one through four. These positions are the default positions.

You will now be provided with a listing of your entries as shown in Figure 1.15. Tear your printout off and compare your listing to that figure.

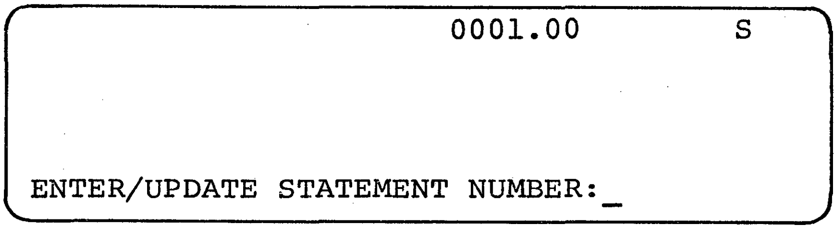
```
QINVEN

0001 FINVENT  IPE    52  52  6AI     2 DISK
0002 IINVENT  AA  01    1 CI
0003 I                                       2   70ITNUM
0004 I                                       8   90CLASS
0005 I                                      10  21 DESC
0006 I                                      22  262PRICE
0007 I                                      27  310ONHAND
0008 I                                      32  360ONORD
0009 I                                      37  410BKORD
0010 I                                      42  460SLSMTD
0011 I                                      47  510SLSYTD
0012 I                                      52  52 DCODE
```

Figure 1.15

Make sure all entries are identical. If there are any changes to be made, we can make them at this time.

As an exercise, let's show you how to make a change even though all your entries were correct.

Re-key the command SEU QINVEN,R now and press the Enter button. The resulting screen will be identical to Figure 1.16.



```
                        0001.00          S



ENTER/UPDATE STATEMENT NUMBER:_
```

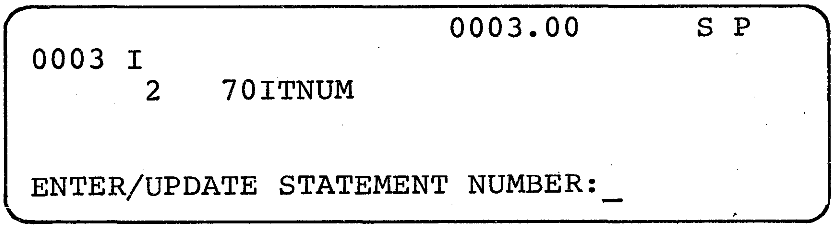
Figure 1.16

Notice that it is different than the original screen obtained when entering SEU. In this case, as QINVEN is already stored, SEU assumes that we will not be entering QINVEN, but rather updating it. Therefore, the screen only asks for the statement number we wish to update or change.

Now press Command Key 6 so that we will print our change.

Let's change the second line, line number 2, which is the first Input specification line in Figure 1.15. Suppose we wanted to change the two alphabetic characters in positions 15 and 16 from AA to BB. Enter a 2 for statement number 2 as that is the number of that line as shown in our listing. The number 0002 appears next to that specification because we chose a serialization option at the end of making our entries. As you can see, serialization can be very helpful when updating files.

Press Field Advance twice until the cursor appears at position 15, type in BB, press the Enter key, and then press Record Advance indicating that that was the only change on that line. Notice that the change has been printed and that the screen is now identical to Figure 1.17.



```
                          0003.00       S P
0003 I
      2    70ITNUM


ENTER/UPDATE STATEMENT NUMBER:_
```

Figure 1.17

When updating with SEU, the next sequential line of coding will appear after any line has been updated.

If we would wish to make any more changes, we would simply key in the statement number of the next statement to be changed and make the appropriate correction.

***

NOTE: Leave the change we have just made as it is.

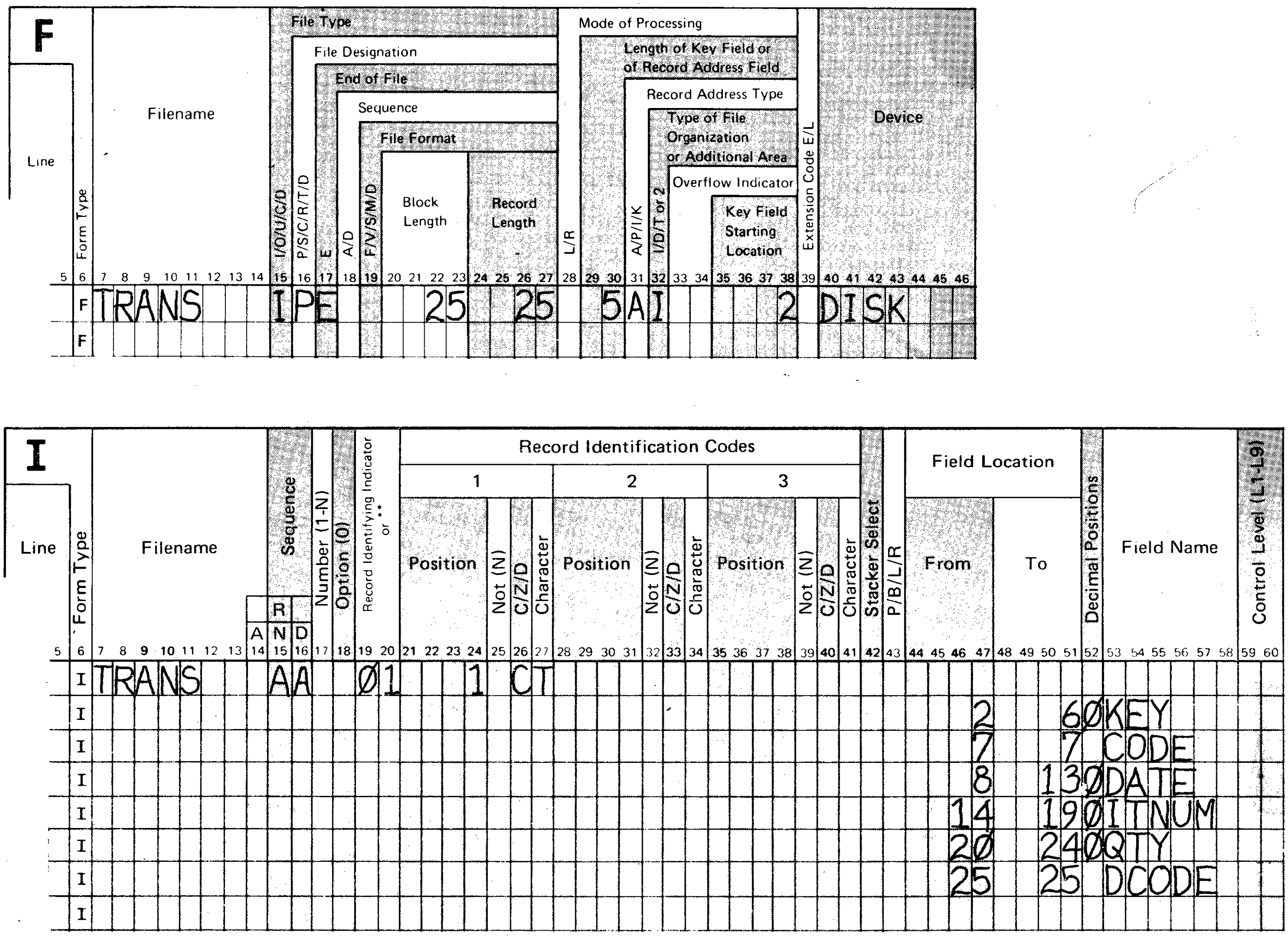
***

After the last change has been made, press Command Key 7 to indicate an end of job.

Again, the same end of job options are made available. As no extra statements have been entered nor old statements deleted, there is no need to reserialize. Take a two option to indicate that a listing is desired.

Now that we have entered QINVEN, we must also enter a source module name for the transaction file. It will be called QTRANS.

The entries for QTRANS are displayed in Figure 1.18.



F
Line
Form Type
Filename
File Type
File Designation
End of File
Sequence
File Format
Block Length
Record Length
Mode of Processing
Length of Key Field or of Record Address Field
Record Address Type
Type of File Organization or Additional Area
Overflow Indicator
Key Field Starting Location
Extension Code E/L
Device
I
Line
Form Type
Filename
Sequence
Number (1-N)
Option (0)
Record Identifying Indicator or **
Record Identification Codes
Position
Not (N)
C/Z/D
Character
Stacker Select
P/B/L/R
Field Location
From
To
Decimal Positions
Field Name
Control Level (L1-L9)

```
FTRANS   IPE    25  25  5AI     2 DISK
F
```

```
ITRANS   AA  Ø1   1 CT
I                                        2   6ØKEY
I                                        7   7 CODE
I                                        8  13ØDATE
I                                       14  19ØITNUM
I                                       2Ø  24ØQTY
I                                       25  25 DCODE
I
```

Figure 1.18

It is an RPG II representation of the structure of the transaction file and the records therein as shown in Figure 1.19.

TRANSACTION FILE

FIELD NAME	POSITION	DECIMALS	FIELD DESCRIPTION
'T'	1	Alpha	Record Identification Code
KEY	2-6	0	DFU Generated Key
CODE	7	Alpha	Transaction Code (R,I or A) (R=Receipt, I=Issue, A=Adjustment)
DATE	8-13	0	Date of Transaction
ITNUM	14-19	0	Item Number
QTY	20-24	0	Transaction Quantity
DCODE	25	Alpha	Delete Code

Figure 1.19

In order to enter QTRANS, enter the SEU command, SEU QTRANS,R and press the Enter button.

When the Keyboard Ready light comes back on (10-15 seconds), press Command Key 3, key the letter F to indicate a File Description specification, and press the Enter button. We are now ready to make our first entry.

Now, after pressing Command Key 6 to print our entries, enter the File Description line. Again, check the screen to make sure that we have made no errors. If so, press the Record Advance key.

Now, depress Command Key 3, the letter "I" and the Enter button to indicate that we are now about to enter the first Input specification. Key only the first Input line as shown in Figure 1.18.

Press Command Key 3 again, the letter "J" and Enter to ready the machine for the entry of field input information. Now enter the field input lines.

After you have entered the last line (for DCODE), again press Command Key 7 indicating an end of job. Take the 4 option for listing and serialization, take the default again for SERIAL START POSITION and compare your list with Figure 1.20.

QTRANS

```
0001 FTRANS   IPE    25  25  5AI     2 DISK
0002 ITRANS   AA  01    1 CT
0003 I                                        2   60KEY
0004 I                                        7   7 CODE
0005 I                                        8  130DATE
0006 I                                       14  190ITNUM
0007 I                                       20  240QTY
0008 I                                       25  25 DCODE
```

Figure 1.20

If there are any changes to be made at this time, re-enter the SEU command with the QTRANS parameter and a comma and "R". Enter the statement number of any statement you wish to change, make the corrections, and when you finish, go to the end of job (Command Key 7) and take option 4. Now we are ready to begin our DFU exercises.

***

NOTE: If you will be taking this course in blocks of time (rather than one continuous session), care should be taken to ensure that the source members just created (QINVEN and QTRANS) are not deleted from the system if it will be used between your sessions.

If this is not possible, the two members should be stored temporarily on a diskette which would then be removed from the system. They would be reloaded again from the diskette when you return to the system.

If the two source members are inadvertently lost, they would have to be created again by repeating this chapter.

In addition, the two data files which will be created in Exercises 1 and 2 must be similarly protected.

***

THIS PAGE LEFT INTENTIONALLY BLANK

EXERCISE 1 - ENTER FUNCTION

INVENTORY MASTER FILE LOAD

There are four functions of DFU:

1) ENTER - To create data files.

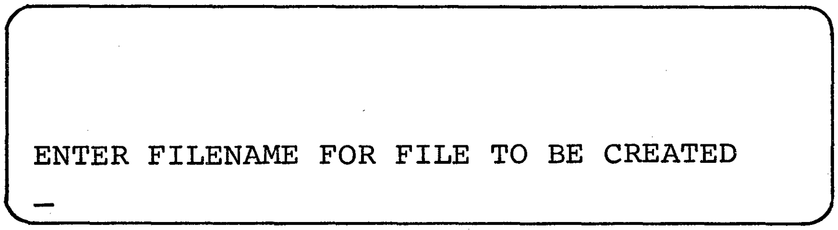
2) UPDATE - To maintain data files (modifying, adding and deleting information).

3) INQUIRY - To display information from data files.

4) LIST - To prepare and print reports from information in data files.

The first function of DFU we will be using is the ENTER function. In this case, we will be loading the inventory master file. All instructions as to how data will be loaded is entered through a series of responses to prompts which DFU will provide on the screen.

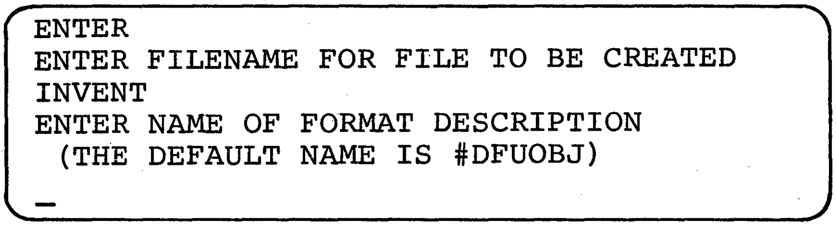
In order to begin our operations, key the word ENTER and then press the Enter button. This will begin the series of prompts. On the screen should be a prompt asking for the file name of the file to be created as illustrated in Figure 2.1.



```
ENTER FILENAME FOR FILE TO BE CREATED
_
```

Figure 2.1

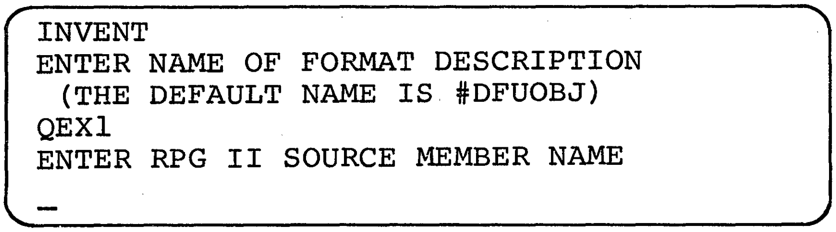
We will call our file name INVENT. (The file name is arbitrarily chosen and usually describes the function of the file.) Key this in and press the Enter button. The next prompt asks for the name of the format description as shown in Figure 2.2.



```
ENTER
ENTER FILENAME FOR FILE TO BE CREATED
INVENT
ENTER NAME OF FORMAT DESCRIPTION
 (THE DEFAULT NAME IS #DFUOBJ)

_
```

Figure 2.2

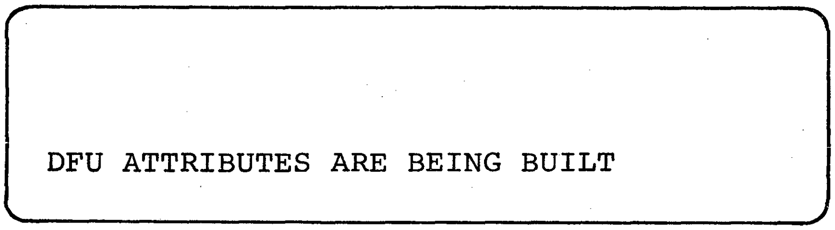
The format description will be the name under which the results of all our prompts will be stored. Our responses will have been translated into machine instructions which will perform the DFU program we are about to describe. If later on we want to use this program again, we will simply refer to it by this name. At this point, key in QEX1 and press the Enter button. The screen should have a display identical to Figure 2.3.



```
INVENT
ENTER NAME OF FORMAT DESCRIPTION
 (THE DEFAULT NAME IS #DFUOBJ)
QEX1
ENTER RPG II SOURCE MEMBER NAME

_
```

Figure 2.3

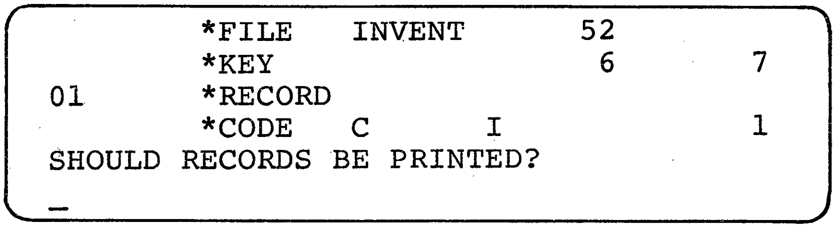
The next response as requested from the screen is illustrated in Figure 2.3. The RPG source member name is asked for. This is the member name which we have previously loaded using SEU. If you remember, the name for the source member representing the inventory master file is QINVEN. This should be entered now. After QINVEN is keyed, the statement 'DFU ATTRIBUTES ARE BEING BUILT' is temporarily displayed on the screen as shown in Figure 2.4.



```




 DFU ATTRIBUTES ARE BEING BUILT

```

Figure 2.4

Soon the screen shows a display as illustrated in Figure 2.5.



```
          *FILE    INVENT       52
          *KEY                   6        7
01        *RECORD
          *CODE    C       I              1
SHOULD RECORDS BE PRINTED?

_
```

Figure 2.5

The first four lines are the beginning of what is called DFU attributes for the inventory master file. This information was taken from the source member we loaded called QINVEN. DFU attributes will be explained later in the course.

The first prompt appears on line 5 and asks "SHOULD RECORDS BE PRINTED?". Such a prompt is answered by a yes or a no. A yes response is indicated by the depression of Command Key 9. A no response is indicated by the depression of Command Key 10 (the zero key immediately to the right of Command Key 9).

Take a look at Figure 2.6 which illustrates the assignment of all the command keys when loading a file through DFU.

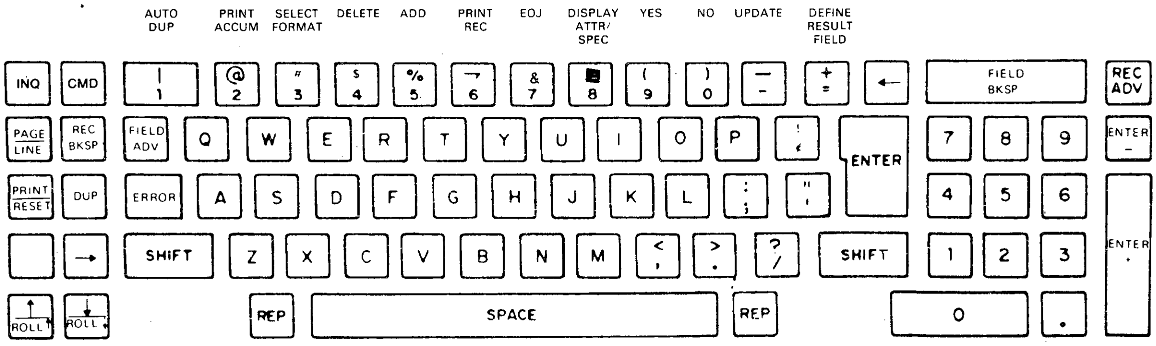


Figure 2.6

Notice that Command Key 9 is labeled YES, and Command Key 10 is labeled NO. These two command keys will be used to answer all yes or no type prompts. (Typing the words YES or NO is invalid as a response to this type of prompt.)

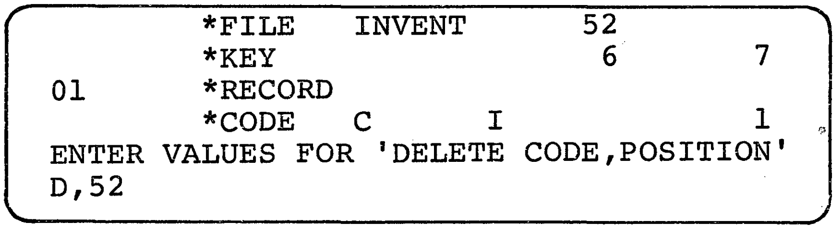
***

NOTE: Consider filling out a keyboard template with the Command Key functions shown above and placing it above the keyboard. This will be helpful for reference throughout the course.

***

The cursor is presently positioned on the sixth line. It is on this line that our response would be typed. In the case of a yes or no answer, there is no need to type onto this line. What we want to do in this case is print out our records as we are loading them so we will answer the question "SHOULD RECORDS BE PRINTED?" with a yes answer by depressing Command Key 9.

Upon depressing that Command Key, another prompt is displayed as illustrated in line 5 of Figure 2.7. It asks for the delete code and position.



```
        *FILE   INVENT      52
        *KEY                 6       7
01      *RECORD
        *CODE   C      I             1
ENTER VALUES FOR 'DELETE CODE,POSITION'
D,52
```

Figure 2.7

This is a field that will be in every record and will be used later on as we update records. For the time being, simply type in D,52 indicating a code D in position 52 as shown in Figure 2.7. Then press the Enter button.

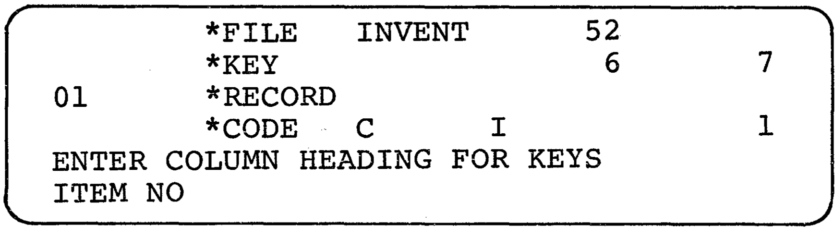
The next prompt asks if all record keys are numeric. In this case, answer yes through the use of Command Key 9. (A key field is that field within a record unique to that record through which that record can be randomly addressed by a program.)

***

NOTE: As you enter the prompts, if you make an incorrect response and have already pressed the Enter button (or Command Key), continue with the job. You will be shown how to make corrections later in the exercise.

***

The next prompt asks for the column heading for these keys. Here, we should enter a literal that will describe the key fields printing immediately beneath it on our printout for the inventory master file load. The key for our inventory master file will be the item number. Therefore, let's key ITEM NO as our column heading as illustrated in Figure 2.8 and press the Enter button.



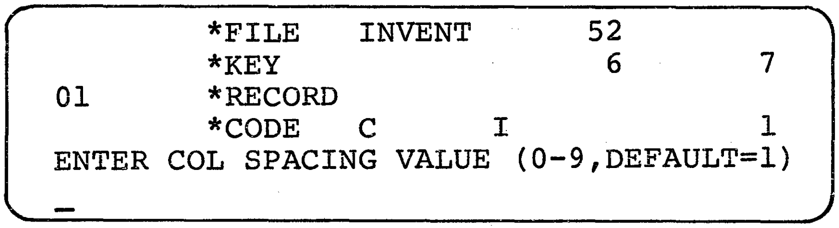
```
          *FILE    INVENT       52
          *KEY                   6       7
01        *RECORD
          *CODE    C      I              1
ENTER COLUMN HEADING FOR KEYS
ITEM NO
```

Figure 2.8

As perhaps you have noticed by now, previous prompts dictate which future prompts will appear. For example, if we had answered no to the question "SHOULD RECORDS BE PRINTED?" this question regarding column headings would not appear now.

The next prompt asks for column spacing values. This pertains to the number of spaces between fields as we will print them out from left to right. Generally, one space would be adequate and would save the most amount of space on our printout. A 1 would almost always be used for those printouts in which the width is an important factor.

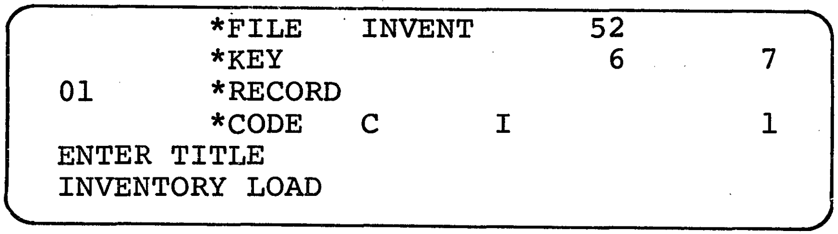
For example, if we had a large number of fields and they might not fit on one line, you might choose one to conserve space. As 1 is the default value as illustrated in Figure 2.9, press the Enter button. There is no need to key a 1 first.



```
          *FILE    INVENT       52
          *KEY                   6       7
01        *RECORD
          *CODE    C      I              1
ENTER COL SPACING VALUE (0-9,DEFAULT=1)

_
```

Figure 2.9

To the next response (ENTER TITLE), key INVENTORY LOAD as illustrated in Figure 2.10 and press the Enter button. Notice again that the last two prompts would not have been displayed unless we have answered yes to the earlier prompt "SHOULD RECORDS BE PRINTED?".

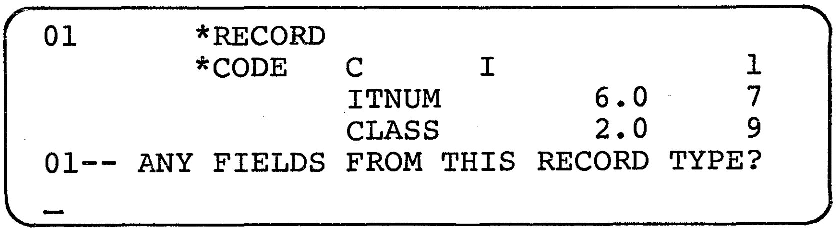


```
         *FILE   INVENT       52
         *KEY                  6       7
01       *RECORD
         *CODE   C      I              1
ENTER TITLE
INVENTORY LOAD
```

Figure 2.10

For the first time since the prompting began, the first four lines of our screen have changed as shown in Figure 2.11. Instead of referring to the inventory file as a whole, the four lines now displayed refer to a record within that file.

Within the source member QINVEN that we defined earlier through SEU, we could have defined many different records within that module. These would have been indicated by different record input lines. If you remember, our first record input line indicated that all our records had the character "I" in position 1. Other records could have been defined within QINVEN with other identifying characters. However, in our case, we only used one type of record, and this record is the one that is displayed now on the screen.



```
01       *RECORD
         *CODE   C      I              1
                 ITNUM         6.0     7
                 CLASS         2.0     9
01-- ANY FIELDS FROM THIS RECORD TYPE?

_
```

Figure 2.11

The prompt on line 5 asks if there are any fields from this record type. In this case, the answer is yes - the record type being shown is the only record type within the file which will be called INVENT. Therefore, use Command Key 9 to indicate a yes response to this question.

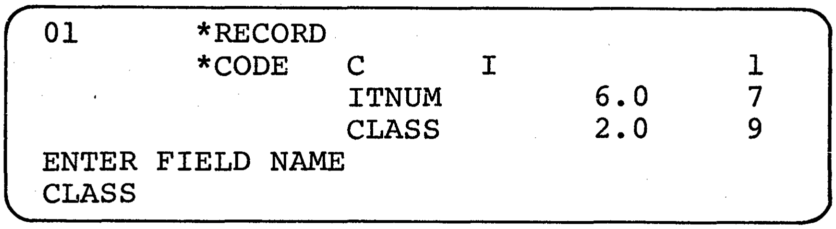
The next prompt asks us to enter field name. At this point, we will begin entering all the fields that were described in our source member and which are to be loaded during this run. Take a look at Figure 2.12 which is a duplicate of our SEU entry for QINVEN.

```
QINVEN

0001 FINVENT   IPE     52  52  6AI      2 DISK
0002 IINVENT   AA  01    1 CI
0003 I                                           2   70ITNUM
0004 I                                           8   90CLASS
0005 I                                          10  21 DESC
0006 I                                          22  262PRICE
0007 I                                          27  310ONHAND
0008 I                                          32  360ONORD
0009 I                                          37  410BKORD
0010 I                                          42  460SLSMTD
0011 I                                          47  510SLSYTD
0012 I                                          52  52 DCODE
```

Figure 2.12

This lists all of our fields. Now we must enter the fields described therein. The exception to this is the first field, ITNUM (the item number). This has already been defined earlier in our prompts when we described our key. Furthermore, information on the "F" line of QINVEN indicates that our key is in positions 2 to 7 where ITNUM is described. Therefore, it does not have to be further described as DFU knows where that field is and as we have already indicated the heading for that field (ITEM NO) in response to a previous prompt.

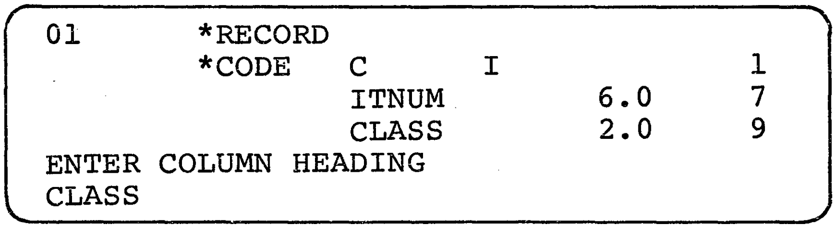
Therefore, the first field we will enter is the second field from QINVEN - CLASS. Type that in now as illustrated in Figure 2.13. Then press the Enter key.



```
01       *RECORD
         *CODE    C       I             1
                  ITNUM          6.0    7
                  CLASS          2.0    9
ENTER FIELD NAME
CLASS
```

Figure 2.13

The next prompt asks for the column heading we wish to print above this field as it is loaded. As the name CLASS, which is the field name, is also appropriate for our heading, let's enter CLASS again for the column heading as shown in Figure 2.14.



```
01      *RECORD
        *CODE   C       I               1
                ITNUM          6.0      7
                CLASS          2.0      9
ENTER COLUMN HEADING
CLASS
```

Figure 2.14

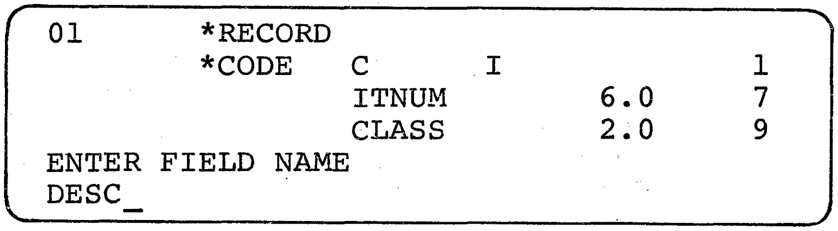
The next prompt asks if this field (CLASS) is an auto-dup field. An auto-dup field is one that would automatically be duplicated by DFU from record to record as they are loaded. This saves on repetitive keying. It might be done, for example, if we were loading a series of records which all have the same date. In that case, that field would be specified an auto-dup field to save keying time as we are loading the file. In this case, we are describing CLASS, and CLASS will vary from record to record. So therefore, we should not describe this field as auto-dup and respond through Command Key 10 with a no response.

Upon doing this, the next prompt asks the question, "ACCUMULATE THIS FIELD?". Fields are accumulated during file loads to give totals which may be used as a control against a separate calculation of such totals. Often if a field being described is an amount field, these amounts should be accumulated so that at the end of the job we may match the totals of the quantities that we have keyed against another total that was taken off line before we began the job. This helps to verify that the entries we made were correct.

While theoretically every numerical field to be described could be accumulated, there is a point beyond which extra controls are usually not necessary. In the case here, we will be indicating the accumulation of five fields for the record. CLASS will not be one. Therefore, respond no to this prompt by using Command Key 10.

This in turn, prompts the next question asking "IS THIS A SELF-CHECK FIELD?". Self-checking is a technique that is sometimes used to ensure the accuracy of control numbers in data records. They will not be used during this file load or during this course. For now, answer no to this prompt using Command Key 10.

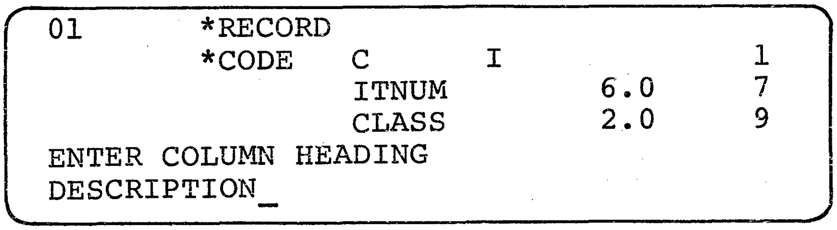
We have now fully defined the field CLASS and answered all prompts associated with it. The next prompt asks for the next field name to be entered. At this point, looking back to our listing for QINVEN (Figure 2.12), we can see that it is DESC. Enter this field now as shown in Figure 2.15.



```
01       *RECORD
         *CODE    C       I                 1
                  ITNUM            6.0      7
                  CLASS            2.0      9
ENTER FIELD NAME
DESC_
```

Figure 2.15

In response to the 'ENTER COLUMN HEADING' prompt, type in the word DESCRIPTION as shown in Figure 2.16, but do not press the Enter key.



```
01       *RECORD
         *CODE    C       I                 1
                  ITNUM            6.0      7
                  CLASS            2.0      9
ENTER COLUMN HEADING
DESCRIPTION_
```

Figure 2.16

By now, you might have deduced that the same five prompts will be repeated for each field to be described. However, in certain cases, there are short cuts which avoid the time consuming entry of all five responses. After the column heading prompt, the next three prompts would ask:

1. Is this an auto-dup field?

2. Should we accumulate it?

3. Is this a self-check field?

If the answer to all three of these questions would be no, we can end the prompting for the description field at this point by pressing the Record Advance key instead of the Enter key. Do this now. The default for all three responses is no, and if you notice now, the next prompt is 'ENTER FIELD NAME' and the System/32 is waiting for the third field to be described.

Similarly, an even faster short cut may be taken in the case of the next field to be entered, PRICE. DFU will use the field name as the column heading as a default unless we indicate otherwise. In this case, the heading for PRICE could be identical to the field name. Again, since PRICE will neither be auto-dup, accumulated, nor self-checking, we do not need to enter any responses other than the field name. Therefore, simply key in PRICE as our field name and press Record Advance, indicating we wish to make no more responses. The field name PRICE will also be the heading for that field as it prints out on our inventory load listing.

The next field to be described is ONHAND. Let's key that now (typing ONHAND as written in our source member - QINVEN) and press the Enter button. We wish to make the heading two separate words and must describe it separately. Therefore, we pressed the Enter button. This caused the column heading prompt to immediately follow. Type in ON HAND with a space between the two words, as this will be heading which we will use. Then press Enter.

For ONHAND (and for the next five fields) we will want to accumulate to obtain control totals to be checked against an offline total. Therefore, we would want to answer yes to the question "IS THIS FIELD ACCUMULATED?". Therefore, to get to that prompt, we pressed Enter (and not Record Advance). Respond no (Command Key 10) to the auto-dup field question, yes (Command Key 9) to the "ACCUMULATE THIS FIELD?" prompt and no to the self-check prompt. Enter the next four fields in the same way as illustrated in Figure 2.17.

ENTER	ENTER	AUTO-DUP	ACCUMULATE	SELF-CHECK
FIELD NAME	COLUMN HEADING	FIELD?	THIS FIELD?	FIELD?
ONORD	ON ORDER	NO	YES	NO
BKORD	BACK ORDER	NO	YES	NO
SLSMTD	SALES M-T-D	NO	YES	NO
SLSYTD	SALES Y-T-D	NO	YES	NO

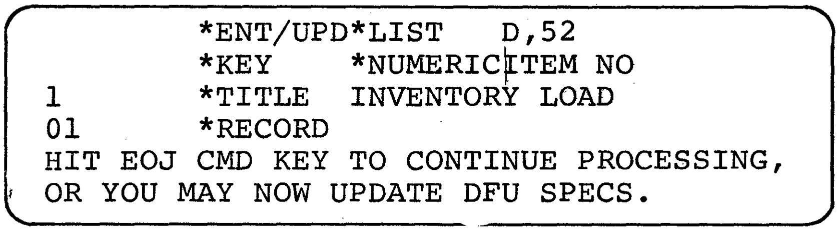
Figure 2.17

Now let's enter the last field - DCODE. Type in DCODE and press the Enter button. In response to the column heading prompt, type the word DELETE and press Enter. Now, in response to the auto-dup field prompt, say yes by pressing Command Key 9.

When loading this file, all delete codes will be blank for each record. Therefore a yes response here will save the operator the time of keying in a blank every time that field value is asked for.

"ACCUMULATE THIS FIELD?" is answered no. The following prompt is "IS THIS A SELF-CHECK FIELD?". Answer no to this prompt.

The next prompt asks for another field name. However, as we have entered all our field names, simply press the Enter button indicating that there are no more fields to be entered. This indicates to DFU that we are at the end of our entries. The display screen will look like Figure 2.18.



```
          *ENT/UPD*LIST   D,52
          *KEY    *NUMERICITEM NO
1         *TITLE  INVENTORY LOAD
01        *RECORD
HIT EOJ CMD KEY TO CONTINUE PROCESSING,
OR YOU MAY NOW UPDATE DFU SPECS.
```

Figure 2.18

What appears on the first four lines are DFU specifications. These are machine translations of the entries we have just been making and will be used in actually loading the file. Lines 5 and 6 display the message to hit the EOJ command key to continue processing or to update the DFU specifications.

If you want to correct an incorrect response made earlier, refer to the DFU Reference Manual (SC21-7600), Chapter 10. Read the three sections on changing, adding and deleting DFU specifications. This will show you how to correct the error. We will demonstrate changing errors in a later exercise.

After you have made corrections, if any, press the EOJ key to continue processing. Referring back to Figure 2.6 we can see that Command Key 7 is labeled EOJ (End of Job). Press it now.

Momentarily displayed on the screen is the message 'DFU DIAGNOSTICS ARE NOW IN PROGRESS'. What is happening now is that all our entries are being checked for conformity with DFU rules. If there are errors detected, the screen will begin flashing and a line will be indicated where an error has been made.

If an error should occur, refer to the Message Guide - Utilities Program Product (SC21-7618) for an explanation of the error. Then refer to the DFU Reference Manual (SC21-7600), Chapter 10. Read the section titled 'Checking the DFU Specifications for Errors'. This will show you how to correct the error.

After the error, if any, has been corrected, DFU will continue to check for yet other errors.

After any errors have been eliminated, DFU assumes that we will immediately begin running the job. In this case, the system is being readied for loading our file. A prompt will appear which asks for the number of records in the file. For our purposes, this will be a small file. Key the number 15 and press Enter. Upon doing this, we are now ready to load our inventory file. Take a look at Figure 2.19. It is a listing of the file we are about to load.

INVENTORY MASTER FILE

REC	ITEM		ITEM		ON	ON	BACK	SALES	SALES	DELETE
CODE	NO	CLASS	DESCRIPTION	PRICE	HAND	ORDER	ORDER	M-T-D	Y-T-D	CODE
(1)	(2-7)	(8-9)	(10-21)	(22-26)	(27-31)	(32-36)	(37-41)	(42-46)	(47-51)	(52)
I	123456	1	WRENCH	8.00	100			100	300	
I	123457	4	NAILS	.50	400		50	25	300	
I	123458	3	BRUSH	2.50	25	500			50	
I	123459	2	NUTS	.50	400	1000		25	25	
I	123460	2	SCREWDRIVER	4.00	100			100	500	
I	123461	3	BOLTS	.75	400			10	10	
I	123462	1	PLIERS	4.25	100	1000		75	150	
I	123463	2	SCREWS	.30	200	500		100	600	
I	123464	1	HAMMER	3.00	50	500		150	500	
I	123465	1	SAW	10.00	40	500		25	550	

Figure 2.19

We will load the ten records described thereon, keying from left to right, Item Number, Class, Description, Price, On-Hand, On-Order and Back Order amounts, and the Sales Month-to-date and Year-to-date amounts.

***

NOTE: Sometimes an error will not become obvious until we are actually running our DFU program. Such an error might be one where we responded not to accumulate a field when we really intended to (that is, we pressed Command Key 10 (NO) when we should have pressed Command Key 9 (YES)). In such a case, we might not have realized we made an error at the time and DFU would not have detected that it was an error.

As we would have already begun running the program, it would be too late to update our specifications. The prompting sequence would have to be repeated and the correct responses made.

If you plan to re-enter specifications for a previously entered program, that program must be deleted first. To do this, enter the command:

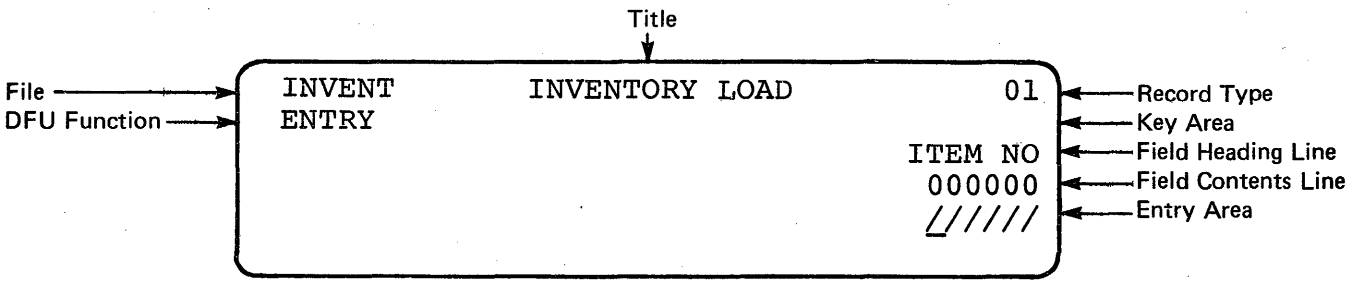
```
REMOVE QEXn,LOAD
```

where n is the number of the exercise.

Later, we will show you a way to store DFU specifications so that they may be modified even after the program has been run. The prompting sequence would not have to be repeated.

***

The "I" in position 1 is a constant factor which was described in our source member and will be written automatically into position 1 of each record. Therefore, our keying will begin with Item number. Take a look at the screen. There should be a display similar to Figure 2.20.

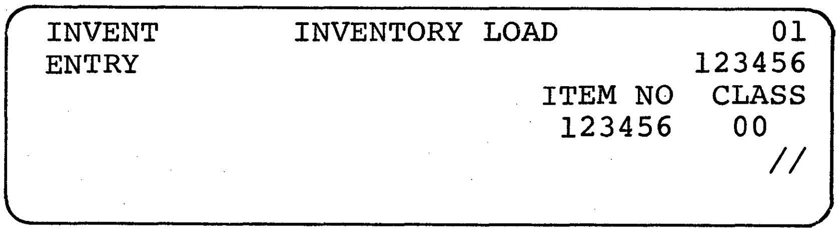


```
INVENT       INVENTORY LOAD          01
ENTRY
                                ITEM NO
                                 000000
                                 //////
```

Figure 2.20

Figure 2.20 describes the function of each of the items displayed, including the title of the report, the name of the file that we are loading, the word ENTRY indicating that we are loading a file, a record type indicated by 01 from our source member, the heading of the first field we are to enter (Item Number), six zeros indicating that it is presently empty, and six slashes - the cursor underneath the first, indicating that this is where we are to enter our Item Number.

Referencing Figure 2.19, key in the first Item Number now, which is 123456, and then press the Enter button. That number is stored under Item Number as you look at your screen (see Figure 2.21).



```
INVENT         INVENTORY LOAD           01
ENTRY                               123456
                         ITEM NO    CLASS
                          123456     00
                                      //
```

Figure 2.21

Notice that as we key each field, it will move to the left across the screen allowing us to visually verify our entry. In addition, the item number appears now immediately under record type (01) and will remain there until we begin to key the next record. This area of the screen holds each record key.

The next field CLASS is requested. In this case, key a 1 and press the Enter button. Similarly, Description is asked for. Key in the word WRENCH and press Enter.

***

NOTE: DFU automatically will right-adjust all numeric fields and position decimals correctly. Do not enter the decimal point (in PRICE) yourself. In addition, if a field about to be entered is zero, simply press the Enter button without making an entry.

***

Now continue keying in the remaining fields until you have keyed Sales year-to-date. DO NOT KEY THE DELETE CODE.

After Sales year-to-date has been keyed in, we come to the delete field which has been described as an auto-dup field. For the first record, of course, we must key in the value which is to be duplicated in succeeding records. In this case, that character is a blank. Therefore press the Enter button without keying an entry.

The screen now requests the Item Number for the next record. Notice that the information from our first record has been printed on the printer. We may visually verify it at this point if we wish by comparing it to the first line in Figure 2.22.

If you detect an error, you will be able to correct it later on. As the next record is entered, you will be shown how to correct a variety of entry errors.

Now, before we continue with the second Item Number, remember that the delete code is an auto-dup field. In order to duplicate this field for all succeeding records and save us time, we must indicate that all fields designated as auto-dup (in this case, just the delete field), should be automatically duplicated. We can do this by depressing Command Key 1, which is our auto-dup key. Do this now.

Notice the characters A and 01 have appeared on the top line of our screen. The A indicates that we are in auto-dup mode duplicating those fields that we described as auto-dup fields during the entry of our specifications. The 01 indicates the record type of the last record which was processed. If that field was different from the present record field as described immediately to its right (which is also 01), we would not want to auto-dup at that time. Instead we would want to change the format of the entry about to be made.

More specifics on how this function would operate is described in the DFU Reference Manual. For now, since all our record types within the file INVENT are the same, we do not have to concern ourselves with that area. Suffice it to say that the auto-dup indicator is on, and all records will be an 01 record type.

Now that we are about to enter our next record, that is, NAILS (Item Number 123457), let's pause for a moment. First, let's now correct some of the errors that might be made upon entry. DFU is now waiting for you to key in the Item Number for the next item, NAILS (123457). Let's say we accidentally hit an alphabetic character at this point. Let's do that now.

Notice the screen immediately starts flashing. We have defined this field as strictly numeric. Press the Error Reset key to stop the flashing and enter the correct item number - 123457.

The next entry needed is CLASS. The class for NAILS is supposed to be 4. Let's key a 5, but do not press the Enter button yet. Press the Field Backspace key. Notice where the cursor is placed (back at the first position). Key in the correct number, 4, and press the Enter button. That was a case where we caught an error before pressing the Enter button.

Now type NAILS and press the Enter button. The next field needed is PRICE. The amount we are to enter is $.50. Instead, key in $1.00 and press the Enter button. (Reminder: Do not key the dollar sign or the decimal point.)

How can we correct this error before we have gone to end of job (where we would use an UPDATE program)? Press The Field Backspace key. That puts us back to PRICE, and again, the system is ready to receive our entry. Let's make the correct entry of $.50 and press the Enter key. Notice that the $.50 which we have now entered replaces the incorrect $1.00 which was previously there.

***

NOTE: The following example is to be used as an illustration only. Do not make this change on the System/32.

***

Similarly, we can retrieve earlier records as long as we have not reached the end of job.

We could change the price of the first item entered (WRENCH - 123456) from $8.00 to $9.00 by keying that item number after we finished keying the present record (NAILS - 123457). After pressing the Enter button, that record would be retrieved and we would then Field Advance to the PRICE field, key $9.00 and press Record Advance. The resulting printout would include the original record reprinted with the new price field appearing immediately underneath the old one.

Errors caught beyond the end of job can be handled through the UPDATE function which we will explain in a later exercise.

For now, let's continue making the entries required for Item Number 123457. Key in the On Hand amount and continue through the Sales year-to-date.

Notice that after entering the Sales year-to-date, we do not stop at delete code. The blank from our first entry has been duplicated automatically into this record, and the line is printed as we have finished our last field entry with Sales year-to-date.

Now proceed entering the rest of the records as described in Figure 2.19. (It is repeated on the next page for reference.)

INVENTORY MASTER FILE

REC	ITEM		ITEM		ON	ON	BACK	SALES	SALES	DELETE
CODE	NO	CLASS	DESCRIPTION	PRICE	HAND	ORDER	ORDER	M-T-D	Y-T-D	CODE
(1)	(2-7)	(8-9)	(10-21)	(22-26)	(27-31)	(32-36)	(37-41)	(42-46)	(47-51)	(52)
I	123456	1	WRENCH	8.00	100			100	300	
I	123457	4	NAILS	.50	400		50	25	300	
I	123458	3	BRUSH	2.50	25	500			50	
I	123459	2	NUTS	.50	400	1000		25	25	
I	123460	2	SCREWDRIVER	4.00	100			100	500	
I	123461	3	BOLTS	.75	400			10	10	
I	123462	1	PLIERS	4.25	100	1000		75	150	
I	123463	2	SCREWS	.30	200	500		100	600	
I	123464	1	HAMMER	3.00	50	500		150	500	
I	123465	1	SAW	10.00	40	500		25	550	

Figure 2.19

Now, since we have entered our last item, we are at an end of job. Depress Command Key 7 to indicate this. Immediately the question, "END OF JOB?" is prompted for us. This is a safeguard to avoid the accidental depression of Command Key 7. If it is not the end of job at this point, we would answer no to this prompt through the use of Command Key 10 (the no key). If it is the end of job, as it is now, Command Key 9 (the yes key), will indicate this. Notice upon depression of Command Key 9, our accumulations have printed out along with the headings that we had entered in response to the prompts. The job is now completed.

Look at Figures 2.22 and 2.23 which should be similar to your printout.

```
07/15/76                              INVENTORY LOAD                              PAGE 001

ITEM NO CLASS DESCRIPTION   PRICE ON HAND ON ORDER BACK ORDER SALES M-T-D SALES Y-T-D DELETE

 123456  01   WRENCH      008.00  00100   00000     00000       00100       00300
 123457  04   NAILS       000.50  00400   00000     00050       00025       00300
 123458  03   BRUSH       002.50  00025   00500     00000       00000       00050
 123459  02   NUTS        000.50  00400   01000     00000       00025       00025
 123460  02   SCREWDRIVER 004.00  00100   00000     00000       00100       00500
 123461  03   BOLTS       000.75  00400   00000     00000       00010       00010
 123462  01   PLIERS      004.25  00100   01000     00000       00075       00150
 123463  02   SCREWS      000.30  00200   00500     00000       00100       00600
 123464  01   HAMMER      003.00  00050   00500     00000       00150       00500
 123465  04   SAW         010.00  00040   00500     00000       00025       00550
```

Figure 2.22

```
                          ON HAND  ON ORDER  BACK ORDER  SALES M-T-D  SALES Y-T-D

BATCH ACCUMULATORS          1815      4000          50          610         2985

TOTAL ACCUMULATORS          1815      4000          50          610         2985
```

Figure 2.23

Notice that all five fields have printed out with the totals of the entries made for those fields. As a matter of fact, they have printed out twice, once on a line labeled 'BATCH ACCUMULATORS', the other on a line labeled 'TOTAL ACCUMULATORS'. In this case, they are both the same as all entries were within one group. Our ten item numbers were consecutive and there was no separate grouping of them. For example, if we had entered our items by class; that is, all class 1 entries first, followed by all class 2, followed by all class 3, etc., we could have had separate accumulators on each class.

If you will notice from referring back to Figure 2.6, Command Key 2 has the function of print accumulators. It would have been used in the following manner. After all four class 1 entries had been entered, we would have pressed Command Key 2. This would have printed out the line BATCH ACCUMULATORS with the appropriate accumulated quantities for just that class. After printing, the accumulators would be reset to zero to begin accumulating from there for the next class or class 2. We would thereby have five totals for each of four classes, plus a final total labeled, TOTAL ACCUMULATORS. If we had used this technique (that is, if we had entered the items in class sequence), the final two lines at end of job would have read, BATCH ACCUMULATORS with just those totals for class 4 followed by TOTAL ACCUMULATORS which would be the accumulators for all entries as is actually displayed now.

At this time, we are at an end of job and would want to check our totals against the off line totals that were kept to ensure the accuracy of our entries. (In this exercise, check your totals against those in Figure 2.23.) If there were any errors, these can be corrected using the UPDATE function of DFU which will be explained later on.

THIS PAGE LEFT INTENTIONALLY BLANK

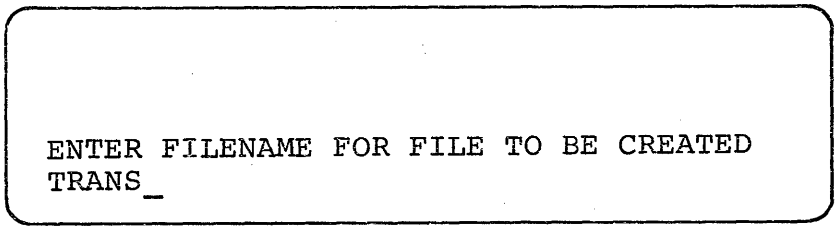
EXERCISE 2 - ENTER FUNCTION

TRANSACTION FILE LOAD

The second DFU exercise will again be using the ENTER function to load to Transaction file. We will follow operations similar to Exercise 1 when we loaded the Inventory Master File.

As in the case of loading the Inventory Master File, a source member has already been loaded which defines the layout of the record in the file. In this case, it was called QTRANS, whereas for the Inventory Master file it was called QINVEN.

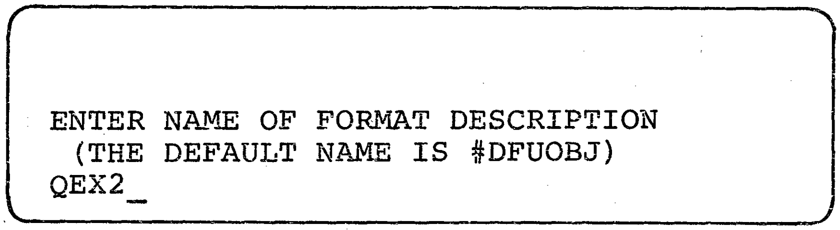
Let's begin the operations as we did in Exercise 1 by typing the word ENTER and pressing the Enter button. As shown in Figure 3.1, the screen prompts us now asking for the file name of the file to be created. Enter TRANS for the name of this file and press Enter.



```
ENTER FILENAME FOR FILE TO BE CREATED
TRANS_
```

Figure 3.1

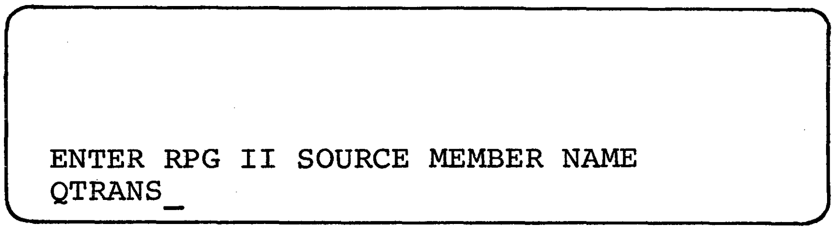
Next, the name of the format description for our entries is requested as shown in Figure 3.2. At this point type in QEX2 for Exercise 2.



```
ENTER NAME OF FORMAT DESCRIPTION
 (THE DEFAULT NAME IS #DFUOBJ)
QEX2_
```

Figure 3.2

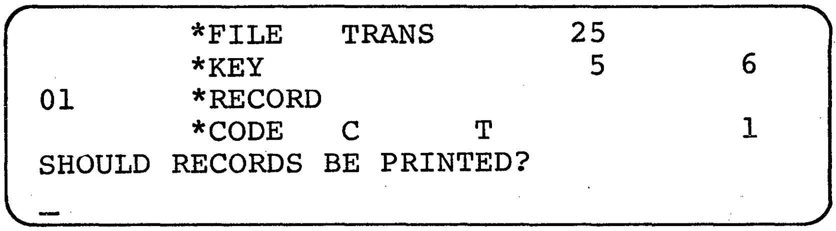
Next, the RPG II source member name describing this file is asked for. As mentioned earlier, this is QTRANS. Key that in now as shown in Figure 3.3.



```
ENTER RPG II SOURCE MEMBER NAME
QTRANS_
```

Figure 3.3

Upon pressing the Enter key, the message DFU ATTRIBUTES ARE BEING BUILT flashes on the screen. When that message disappears, the first of our series of prompts appears. Look at Figure 3.4.



```
         *FILE   TRANS       25
         *KEY                 5       6
01       *RECORD
         *CODE   C     T              1
SHOULD RECORDS BE PRINTED?

_
```

Figure 3.4

As in the case of Exercise 1, the first four lines of the screen identify the. file to be created (Transaction File) and how we set it up in our source member QTRANS. These four lines will remain constant on the screen while we are keying in data for this record. The fifth line on the screen gives us our first prompt "SHOULD RECORDS BE PRINTED?". Using Command Key 9, respond with a yes to this prompt.

Again, the value for delete code and position is requested with the next prompt. As in the case of the Inventory Master File, our delete code will be a D, but it will be in position 25 of our record. Therefore, key in D,25.

The next prompt is new to us. "SHOULD DFU GENERATE KEYS FOR YOU?".

If we want DFU to generate keys for us, a key field of five positions should be specified in our source member. Then when this prompt appears, we would answer yes to it through the use of Command Key 9. In the case of the Transaction file, we do want DFU to generate the record keys for us. Therefore, use Command Key 9 now to answer yes to the question "SHOULD DFU GENERATE RECORD KEYS FOR YOU?".

The next question asks "ENTER COLUMN HEADINGS FOR KEYS". At this point, we could key in a descriptive heading for the key. However, in this case, since DFU will generate a series of sequential numbers which will become the key, there is no descriptive field which might be used to adequately describe it. *KEY is the default for this response. This will be appropriate for our printout. Therefore, simply press the Enter button to obtain *KEY as our heading for the keys.

Column spacing value appears next as was the case in Exercise 1. Key a 2.

The next prompt asks for the title of the report. Let's enter TRANSACTION LOAD.

Again, the prompt "ANY FIELDS FROM THIS RECORD TYPE?" appears as was the case in Exercise 1. Again, since this record type is the only record type described within QTRANS, answer yes to this prompt. Now we begin to enter the succession of field names and descriptive information regarding them. Take a look at Figure 3.5.

```
QTRANS

0001 FTRANS   IPE    25  25  5AI     2 DISK
0002 ITRANS   AA  01    I CT
0003 I                                        2   60KEY
0004 I                                        7   7 CODE
0005 I                                        8  130DATE
0006 I                                       14  190ITNUM
0007 I                                       20  240QTY
0008 I                                       25  25 DCODE
```

Figure 3.5

This is a duplicate copy of QTRANS which was entered earlier. It contains the field names that we wish to key at this point. The key has already been described through previous responses. The code in position 1, a T, is indicated for every record and therefore it is not asked for as we key now. The T will automatically be entered into position 1 of each record of the Transaction file.

Therefore, the first field to be defined is the field called CODE. As in the case of Exercise 1, we have a choice of pressing the Enter or the Record Advance button after each entry. In the case of CODE, we will want to ask ourselves if we want a different heading than the name of the variable itself - CODE. In this case - no, as the word CODE is appropriate to describe the field on our listing. The other questions which would be asked are 'Is it an auto-dup field?', 'Is the field to be accumulated?', and 'Is it a self-check field?'. Again, if the answers to all three questions are no, we may skip those prompts. Therefore, after responding with the word CODE, press Record Advance as we need no more prompts to describe this field.

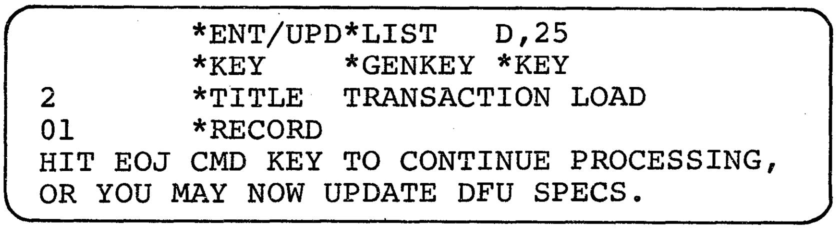
The next field DATE falls into exactly the same situation. DATE is appropriate for the heading. It is not a self-check field. We will not accumulate it, and it is not an auto-dup field. Therefore, simply type in DATE and press the Record Advance Key.

The next field ITNUM (item number) should be typed in next. In this case, we would want to specify a separate heading as the variable name (ITNUM) might not be understood on a listing. Therefore, type in ITNUM, but press the Enter key. The column heading prompt appears next. Key in ITEM NO to respond to this prompt. As the answer to our next three questions would be no, press Record Advance.

The next field is quantity or QTY. Type that and press the Enter button. Our heading for this field should be QUAN. Again, type this and then press the Enter key. We press the Enter key and not Record Advance because we want to accumulate this field. If we pressed Record Advance, the prompt asking if the field should be accumulated would be skipped and the default taken - no, would not allow the field to be accumulated. Therefore the next question which appears is "IS THIS AN AUTO-DUP FIELD?". Use Command Key 10 to signify no. Accumulate this field? - Command Key 9 to signify yes. Is this a self-check field? Command Key 10 to signify no.

Now, we come to the last field to be entered which is DCODE. Type DCODE now and press the Enter button, as we wish to place the heading DELETE over this field. After typing this press the Enter button as we want to have the prompt, "IS THIS AN AUTO-DUP FIELD?" appear. Answer yes to that question, no to the accumulate question, and no to the self-check field question.

Again, the screen requests to enter a field name. As there are no more field names for this record, simply press the Enter key. Now displayed on the screen is Figure 3.6.



```
         *ENT/UPD*LIST   D,25
         *KEY    *GENKEY *KEY
2        *TITLE  TRANSACTION LOAD
01       *RECORD
HIT EOJ CMD KEY TO CONTINUE PROCESSING,
OR YOU MAY NOW UPDATE DFU SPECS.
```

Figure 3.6

The top four lines of our screen have changed from being descriptive of the record to a description of the responses that we have been keying. The four lines which appear are the first four lines of our specifications. The first line indicates that we are doing an ENTER/UPDATE function, that we are listing our records and that the delete code is a D in position 25. The second line indicates that the key is generated by DFU and has a heading of *KEY on the printout. The title of the report is 'TRANSACTION LOAD' as indicated on line 3 as is a spacing value of 2. Line 4 begins the specification of the record that we have described (01).

An option available to us now is to print the entire list of DFU attributes and specifications. We may do this by pressing Command Key 6 (the print command key) at this time. Let's do that now. Look at Figures 3.7 and 3.8.

```
                 ****  DFU ATTRIBUTES   ****

                *FILE    TRANS        25
                *KEY                   5       6
01              *RECORD
                *CODE    C      T              1
                         KEY           5.0     6
                         CODE          1       7
                         DATE          6.0    13
                         ITNUM         6.0    19
                         QTY           5.0    24
                         DCODE         1      25
```

Figure 3.7

```
       ****  DFU SPECIFICATIONS  ****

            *ENT/UPD*LIST   D,25
            *KEY    *GENKEY *KEY
2           *TITLE  TRANSACTION LOAD
01          *RECORD
            *       CODE    CODE
            *       DATE    DATE
            *       ITNUM   ITEM NO
            *ADD    QTY     QUAN
            *D      DCODE   DELETE
```

Figure 3.8

What has printed out is first a list of the DFU attributes which relate to the file as described in QTRANS, except changes have been made to make the format acceptable to DFU. DFU attributes are fully described in Chapter 9 of the DFU Reference Manual.

Secondly, DFU specifications have printed which reflect all the responses we have made to the prompts: which fields to add, what headings to use, what fields to auto-dup, the title of the report, and the position and code for the delete code. DFU specifications are fully described in Chapter 10 of the DFU Reference Manual.

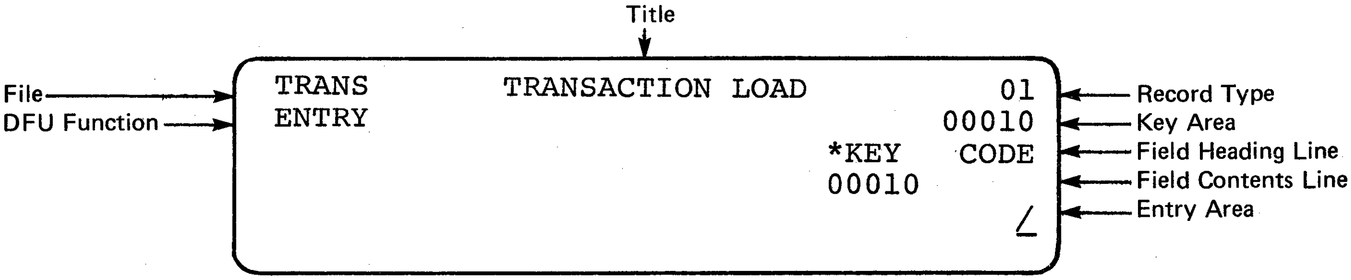
If you notice, there are five more lines which appear on our DFU specifications after those which appeared on the screen. They define the fields that we have described. In the middle column are the five variables' names from the source member QTRANS. In the right-most column is the heading which we will print out over those fields. To the left are the operations, if any, that we will be doing with these fields.

As we indicated, there are no special operations with code, date, or item number. As a matter of fact, code and date have the same heading as variable name while item number has a different heading as we specified. QTY was the field we wish to accumulate and therefore ADD appears to the left on that line. DCODE was the field we wished to auto-dup. The D appearing to the left on that line indicates that this is an auto-dup field.

If we discovered an error at this point and wished to make a change, we can do so directly on these specifications rather than stepping through the entire series of prompts again. In Exercise 3, we will be changing such a specification. For now, let us proceed.

The screen asks us to hit the EOJ key to continue processing or update the DFU specifications. As we do not wish to update, press Command Key 7 to indicate end of job. The screen immediately displays the message 'DFU DIAGNOSTICS ARE NOW IN PROGRESS'. Again, at this point, if there were any errors we hadn't spotted in our specifications, DFU would isolate them at this time and tell us the error that we had made and indicate the line on which that error was made. In Exercise 6 we will correct such an error.

Again the prompt 'ENTER NUMBER OF RECORDS TO BE IN THE FILE' appears. Key 15 and press the Enter button. Look at Figure 3.9.



```
TRANS          TRANSACTION LOAD          01
ENTRY                                 00010
                               *KEY    CODE
                               00010
```

Figure 3.9

It represents the screen as you see it now. Figure 3.9 identifies the function of all those fields which are displayed. Notice as opposed to the display for the Master File during Exercise 1, some differences appear.

1. As the key is automatically generated for this file, it already appears in the key area immediately underneath record type (00010). Keys are automatically numbered 00010, 00020, 00030, etc., when generated by DFU. Also it is already entered into our field contents line. Therefore, the first field to be entered is CODE.

2. The content of CODE is set to a blank (not zero) as it was defined as alphameric and not numeric.

Look at Figure 3.10. It is the data to be loaded into the transaction file.

TRANSACTION FILE

REC CODE (1)	DFU KEY (2-6)	TRANS CODE (7)	TRANS DATE (8-13)	ITEM NO (14-19)	QUAN (20-24)	DELETE CODE (25)
T		I	081575	123456	40	
T		I	080175	123458	10	
T		A	080875	123458	50	
T		R	080975	123458	500	
T		A	081575	123459	100-	
T		A	080875	123460	50-	
T		I	080875	123462	50	
T		R	080175	123463	300	
T		R	081275	123463	200	
T		R	080675	123464	500	
T		R	080175	123465	500	
T		R	080375	123468	100	

Figure 3.10

The System/32 is now ready for the entry of our first record. The first field requested is the code and for the first record the code is an "I". Type "I" and press the Enter button. The next field requested is the date. Key the date now (081575) and press the Enter button. Next, item number is asked for. Key the item number 123456 and press the Enter button. Quantity is asked for next. Key 40 and press the Enter button. The delete code is next. This will be blank as we are now loading these records; later on, if they are to be deleted, a D will be inserted here automatically. However, for the time being press the Enter button without keying an entry to load a blank.

As that was the last field for the first record, we have now printed the record on the printer as we have answered yes to the "SHOULD THESE RECORDS BE LISTED?" prompt. Verify your printout with the first item line in Figure 3.11.

Before we enter the second record, don't forget to press Command Key 1 for the auto-dup function so that delete codes may be duplicated from record to record.

At this point, key in the second record - "I" for code, the date, the item number, and the quantity of ten. Notice that we printed immediately after keying quantity. As the delete code in the next field is now automatically duplicated, we do not have to key it again.

Before entering the remaining records, notice that in the fifth and sixth records the quantity field is minus - a minus 100 for Record 5 and a minus 50 for Record 6. After keying those numbers, press the Enter Minus key to indicate that these fields are negative. Also notice that the key field will be automatically updated by DFU, the first key being 00010, and the key for the last record being 00120.

Now key the rest of the records.

When we have entered the last record, indicate end of job. The prompt appears asking us is this indeed the end of job. Depress Command Key 9 to respond yes to that question. Again, lines for both batch and total accumulators print out and your printout should be identical to Figure 3.11. As we did not use Command Key 2 to indicate any batch totals, the batch accumulator and total accumulator are the same at 2100.

```
        07/15/76           TRANSACTION LOAD         PAGE 001

        *KEY    CODE            DATE   ITEM NO       QUAN    DELETE

        00010     I           081575   123456      00040
        00020     I           080175   123458      00010
        00030     A           080875   123458      00050
        00040     R           080975   123458      00500
        00050     A           081575   123459      00100-
        00060     A           080875   123460      00050-
        00070     I           080875   123462      00050
        00080     R           080175   123463      00300
        00090     R           081275   123463      00200
        00100     R           080675   123464      00500
        00110     R           080175   123465      00500
        00120     R           080375   123468      00100

                                                                  QUAN

    BATCH ACCUMULATORS                                            2100

    TOTAL ACCUMULATORS                                            2100
```

Figure 3.11

THIS PAGE LEFT INTENTIONALLY BLANK

EXERCISE 3 - UPDATE FUNCTION

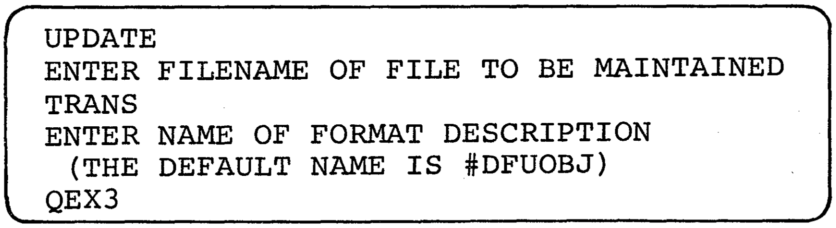
TRANSACTION FILE UPDATE

Exercise 3 will update the Transaction file. All three types of updating will be done, that is, we will add a record to the file, we will delete another record from the file, and we will change a record in the file (that is, changing a field within that record). In addition to performing these operations, we will also experiment in changing DFU specifications after entry and before the end of job.

As this job will be an update, we will key the initial prompt of UPDATE as opposed to ENTER which we had keyed previously. Key UPDATE now.

The screen next asks for the file name of the file to be maintained. In this case it's our transaction file which we loaded under the name TRANS. Enter this now.

The screen prompts for a name of the format description (that is the module where the result of our prompts will reside). Let's call it QEX3 as shown in Figure 4.1.



```
UPDATE
ENTER FILENAME OF FILE TO BE MAINTAINED
TRANS
ENTER NAME OF FORMAT DESCRIPTION
 (THE DEFAULT NAME IS #DFUOBJ)
QEX3
```

Figure 4.1

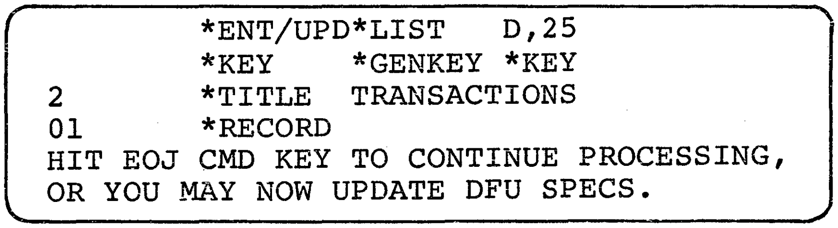
Next, the name of the source member name is asked for. Type QTRANS. After the message 'DFU ATTRIBUTES ARE BEING BUILT' is displayed on the screen our prompts begin.

Respond to the rest of the prompts as indicated on the following page.

PROMPT	RESPONSE
SHOULD RECORDS BE PRINTED?	YES (Command Key 9)
ENTER VALUES FOR 'DELETE CODE, POSITION'	D,25,Enter
SHOULD DFU GENERATE RECORD KEYS FOR YOU?	YES (Command Key 9)
ENTER COLUMN HEADING FOR KEYS	Enter (*KEY is the default)
ENTER COLUMN SPACING VALUE (0-9,DEFAULT=1)	2,Enter
ENTER TITLE	TRANSACTIONS,Enter
01-ANY FIELDS FROM THIS RECORD TYPE?	YES (Command Key 9)
ENTER FIELD NAME	CODE, Record Advance
ENTER FIELD NAME	DATE, Record Advance
ENTER FIELD NAME	ITNUM, Enter
ENTER COLUMN HEADING	ITEM NO, Record Advance
ENTER FIELD NAME	QTY, Enter
ENTER COLUMN HEADING	QUAN, Enter
IS THIS AN AUTO-DUP FIELD?	NO (Command Key 10)
ACCUMULATE THIS FIELD?	YES (Command Key 9)
IS THIS A SELF-CHECK FIELD?	NO (Command Key 10)
ENTER FIELD NAME	DCODE, Enter
ENTER COLUMN HEADING	DELETE, Enter
IS THIS AN AUTO-DUP FIELD?	YES (Command Key 9)
ACCUMULATE THIS FIELD?	NO (Command Key 10)
IS THIS A SELF-CHECK FIELD?	NO (Command Key 10)
ENTER FIELD NAME	Enter

Notice we entered the field names identically to the way we entered them for the ENTER Transaction file program. The field headings were the same as the field name for both code and date, and changed for item number to ITEM NO.

In addition to making the heading for quantity QUAN, we accumulated the field. For delete code, we made it an auto-dup field and made the heading DELETE. Since we have keyed the last field and pressed Enter when DFU requested yet another field name, we have the display shown in Figure 4.2.



```
         *ENT/UPD*LIST   D,25
         *KEY    *GENKEY *KEY
2        *TITLE  TRANSACTIONS
01       *RECORD
HIT EOJ CMD KEY TO CONTINUE PROCESSING,
OR YOU MAY NOW UPDATE DFU SPECS.
```

Figure 4.2

Now, let's update a DFU specification instead of simply pressing Command Key 7 to continue processing. The title was entered as TRANSACTIONS. Let's say this was an error and we really should have entered TRANSACTION UPDATE. How can we change this before the program begins?

As indicated in Exercise 2, our specifications consist of a number of lines, the first four of which are displayed on the screen. The one we would like to change is on the screen and is the third line. First, we must access that line. To do this, press the roll down key once. Notice that the line for title has now moved to the fourth line of the screen. This is the point where we can access it. Changes can be made only to the line that is positioned fourth on the screen.

The cursor is at the left under the number 2. We do not want to change that. Rather we want to move to the right-most field - TRANSACTIONS which is our heading. To do this, press Field Advance twice. Notice the cursor is now under the T of the word TRANSACTIONS. Now, make the entry TRANSACTION UPDATE. Then press the Enter key or Record Advance key.

As this is the last field on the line, it makes no difference since upon depression of the Enter key, we will go to the next line. If we had changed the second field of this line instead of the last, the cursor simply would have skipped to the next field on that same line upon depression of the Enter key.

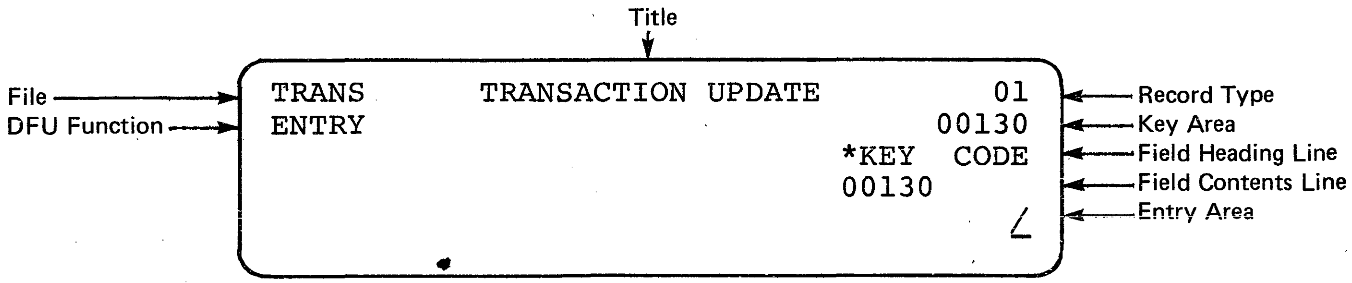
Now that we have changed the heading, print out the DFU specifications by pressing Command Key 6.

It is not necessary to press Command Key 6 to print out the specifications with each job we enter. However, it is good to have a record of our specifications if we may wish to update them later. We will use Command Key 6 during this course to print out the attributes and specifications for reference purposes. Notice that when the specifications printed out the new title, TRANSACTION UPDATE appeared.

Now, press Command Key 7 to indicate the end of job and to begin the actual update. The message 'DFU DIAGNOSTICS ARE NOW IN PROGRESS' appears on the screen. If there are no diagnostics, we will begin execution.

If there are, the line in error will be the fourth line on the screen. Stop the flashing, look at the line and determine what the error is through reference to your System/32 Messages Guide - Utilities Program Product (SC21-7618). Then make the change to continue the job. Complete information on interpreting and changing DFU specifications is addressed in Chapter 10 of the System/32 Utilities Program Product Reference Manual - DFU (SC21-7600).

Now we are ready to begin the actual updating. Notice the screen. It should be identical to Figure 4.3. Explanations of the fields displayed are also shown in Figure 4.3.



```
TRANS       TRANSACTION UPDATE        01
ENTRY                              00130
                              *KEY  CODE
                              00130
                                       ∠
```

Figure 4.3

Notice the key area. It is numbered 00130. If you remember from the transaction load, we loaded twelve records, the last of which was key 00120. When we update, DFU will automatically assume that you will be adding records at the beginning of your run starting immediately after the last record previously entered.

This will not be the case here. While we will add a record, it will appear back in the middle of the file. In order to put DFU into a mode where we may address records selectively, press Command Key 11 (that is, the CMD key with the minus sign). See Figure 4.4.

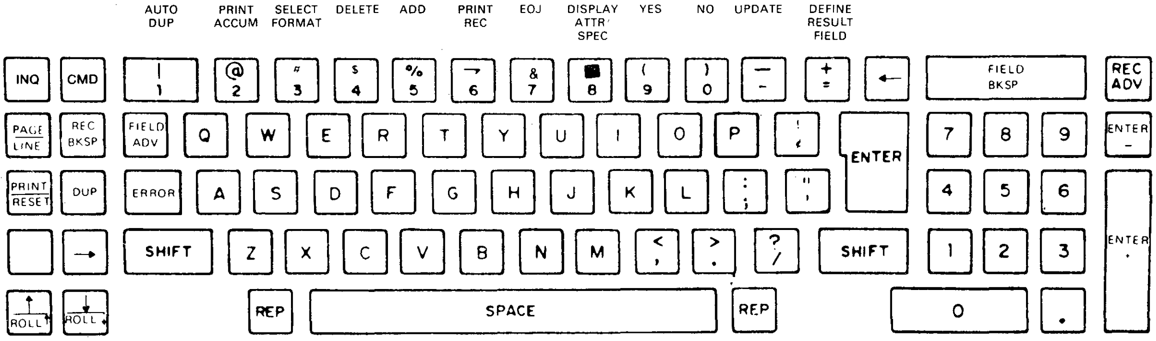
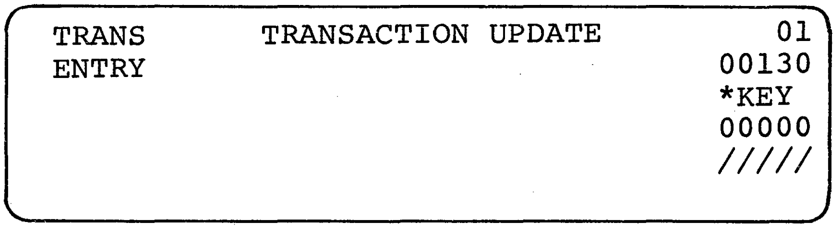


Figure 4.4

Your screen has changed so that it looks like Figure 4.5.



```
TRANS          TRANSACTION UPDATE          01
ENTRY                                   00130
                                         *KEY
                                        00000
                                        /////
```

Figure 4.5

Notice that the field contents line is now five zeros for *KEY and underneath that are five slashes indicating that DFU is waiting to accept the key of the record we wish to address. The first record we wish to address is the sixth record in our file (see Figure 4.6).

TRANSACTION FILE

REC CODE (1)	DFU KEY (2-6)	TRANS CODE (7)	TRANS DATE (8-13)	ITEM NO (14-19)	QUAN (20-24)	DELETE CODE (25)
T	00010	I	081575	123456	40	
T	00020	I	080175	123458	10	
T	00030	A	080875	123458	50	
T	00040	R	080975	123458	500	
T	00050	A	081575	123459	100-	
T	00060	A	080875	123460	50-	
T	00070	I	080875	123462	50	
T	00080	R	080175	123463	300	
T	00090	R	081275	123463	200	
T	00100	R	080675	123464	500	
T	00110	R	080175	123465	500	
T	00120	R	080375	123468	100	

Figure 4.6

This record is key number 00060. The quantity was entered as a minus fifty. It should have been a plus fifty. Let's make that change now.

First, enter the key of this record - 00060. Immediately displayed is the first field which might be changed - CODE. We do not wish to change it. Therefore, press Field Advance. The next field - DATE appears. Again, press Field Advance. Item number appears. Again, we do not wish to change this. Press Field Advance again.

Now we come to the quantity field. Notice that it is minus fifty. Immediately underneath the minus fifty is the minus fifty represented again with the cursor underneath the left-most zero. At that point, key in 50 and press the Enter Plus key to indicate that is a plus field. We do not wish to change the next and last field, the delete field. Therefore, press Record Advance as we have finished making changes to this record. Notice the printout we have obtained as shown in Figure 4.7.

```
07/15/76      TRANSACTION UPDATE        PAGE 001

*KEY    CODE       DATE   ITEM NO       QUAN  DELETE

00060     A     080875   123460       00050-
                                      00050
```

Figure 4.7

***

NOTE: The printout will appear during this program as a result of having answered yes to the 'SHOULD RECORDS BE PRINTED?' prompt.

***

First the report heading has printed out, along with the field headings. A line is printed out representing the old record. Any field that is changed prints out again on a second line immediately under the same field in the first line. (If a field has not been changed, the second line prints out blank for that field.) This way we keep a permanent record of changes that have been made. The quantity for key 00060 has now been changed to plus fifty. This is the first type of change.

The next type of change we will do is to delete a record. Reference Figure 4.6. In this case, we wish to delete the last record - 00120, because it has an invalid item number. There is no record for item number 123468 in the master file and therefore this record was entered in error. We now want to delete it.

Notice we are now waiting for a new key to be entered. Key in 00120 and press the Enter button. We can confirm that this is the record we want to delete by pressing Field Advance a sufficient number of times to check its fields' contents. In this case, field advancing to the item number field and seeing 123468 is sufficient to indicate that this is the record we wish to delete.

Having done that, depress Command Key 4 as shown in Figure 4.4 to delete the record. What prints out is the record as it existed on the file with the exception that now a "D" is positioned in the delete code area as shown in Figure 4.8.

```
07/15/76     TRANSACTION REPORT      PAGE 001

*KEY    CODE      DATE   ITEM NO    QUAN   DELETE

00060     A    080875    123460   00050-
                                  00050
00120     R    080375    123468   00100       D
RECORD DELETED
```

Figure 4.8

The D is placed in position 25 of the record (as we had indicated in our responses to the prompts) and under the heading DELETE. In addition, the statement RECORD DELETED appears immediately underneath it to assure that this record has been deleted during this run.

Later, in any other programs that might access this file, the delete code would be checked and if a D was in position 25 as is now the case, this record would be ignored. During a file maintenance run, all records with this D will be removed from the file.

Finally, the third operation we might wish to do during an update is to add a record. In this case, look at Figure 4.9.

```
*KEY    CODE      DATE   ITEM NO    QUAN   DELETE

00085     I    080275    123463   00250
```

Figure 4.9

This is a record we want to add between the eighth and ninth records in the file. The way to do this is to first create a unique key for it which has a number between the numbers of the keys for the eighth and ninth records (in this case, 00085). Key this in now.

DFU knows that there is presently no such record in our file and thereby knows that we are about to add a record. Code is asked for. Key an I and press Enter. Date is asked for. Key in 080275. Similarly, key item number, quantity and delete code (a blank at this time).

After delete code has been keyed, a line prints out on our report with the keyed data on it. You can tell that it is an addition as neither the D appears under DELETE which would signify a delete, nor two lines appear with a change to a field indicating that a change has been made to that record. Thereby this is truly an addition.

Any entries that were made in error and are discovered at that point can simply be rechanged again as we are in the middle of an update program. For example, if we had decided that item number 123468 should not have been deleted, we would simply key that key back in (00120) and change the delete code back to a blank.

This is the last of our updates. Depress Command Key 7 to indicate end of job. Again, the prompt with the question "END OF JOB?" appears. Answer yes by pressing Command Key 9.

```
07/15/76     TRANSACTION REPORT      PAGE 001

*KEY    CODE      DATE   ITEM NO    QUAN   DELETE

00060     A    080875    123460   00050-
                                  00050
00120     R    080375    123468   00100       D
RECORD DELETED
00085     I    080275    123463   00250


                                                        QUAN

BATCH ACCUMULATORS                                      250

TOTAL ACCUMULATORS                                      250
```

Figure 4.10

Notice again in Figure 4.10 that batch and total accumulators have printed. If we were updating in batches, we could have taken batch totals by pressing Command Key 2 as we were proceeding. However since we did not, the batch accumulator and the total accumulator are the same.

The total 250 in this case indicates the addition of a record with a quantity of 250, the deletion of a record that had a quantity of 100, plus the changing of a quantity field in a third record from a minus fifty to a plus fifty or a plus 100 net giving us a total net change on quantity of a plus 250.

At this point, you should use this program to correct any errors made when you loaded the transaction file. Simply re-enter the UPDATE command and respond to the only two prompts with: TRANS (File Name) and QEX3 (Format Description). No other prompts are required as the format description contains all the responses we made previously.

If we wish to correct any entry errors made when we loaded the Inventory Master file (INVENT), we can use the format description created when we answered the prompts to load that file (QEX1). Enter the UPDATE command then respond to the two succeeding prompts with INVENT (File Name) and QEX1 (Format Description).

***

NOTE: Before creating a format description for updating a file, check how similar the format description that was used to load the file (ENTER) is. If your responses to the prompts would be the same or almost the same as the ENTRY format description for that file, you may well want to use that format description instead of going through a new prompting sequence for updating it.

***

The techniques and benefits of re-using a previously created format description will be fully explained during the next exercise - INQUIRY.

EXERCISE 4 - INQUIRY FUNCTION

INVENTORY MASTER FILE INQUIRY

Exercise 4 will use the third function of DFU - INQUIRY. An inquiry program is one which displays on the screen (or prints) the status of fields from a record selected by the operator. Inquiries into the status of many different records may be done during the running of a single inquiry program.

We will inquire into our inventory master file. What we want displayed on the screen in answer to our inquiry with an item number are its class, description, and on hand, on order, and back order amounts. We may also choose to print the results of our inquiry to have a permanent record of the data retrieved.

An inquiry of this type might be used to quickly determine if there are a sufficient number of units of a particular item on hand to satisfy an order.

To begin the job, type in INQUIRY, and press the Enter button.

A prompt appears asking for the file name of the inquiry file. In this case, key INVENT, the inventory master file.

The next prompt asks for the name of the format description. Key QEX4.

The name of the source member is then requested. This is QINVEN, the source member describing the inventory master file. After keying this and pressing the Enter button, the message 'DFU ATTRIBUTES ARE BEING BUILT' is displayed and after that, we will continue with the series of prompts.

The first prompt which appears is the question, "ARE YOUR RECORD KEYS ALL NUMERIC?". In our case, answer yes (Command Key 9). "ENTER COLUMN HEADING FOR KEYS" is the next prompt. For this file, if you remember, the key was not generated by DFU, and it is the item number field. Therefore, key in ITEM NO for the heading for keys. Now continue with the following responses. COLUMN SPACING VALUE - 2, TITLE - INVENTORY INQUIRY, "ANY FIELDS FROM THIS RECORD TYPE?" - YES (this is the only record type within QINVEN). The next prompt is "ENTER FIELD NAME".

In case of the INQUIRY, the five field-related prompts that we had with the ENTER and UPDATE functions will not be used. Rather only the first two (field name and heading) are prompted during the inquiry. There is no need for the prompts on auto-dup, accumulating, or self-checking to appear.

If you remember, our inquiry is supposed to display, in addition to item number, class, description, on hand, on order, and back order amounts. As we already entered the information for item number when the information for key was being prompted, the next field name to be entered is CLASS. For this exercise, we will show you another technique to enter field names.

You may prefer, in some cases, rather than typing the the word CLASS, to duplicate it from the DFU attributes being displayed on the screen. The field appearing in the fourth line of the screen would be entered as the field name upon pressing the Dup key. For example, in this case, the next field we want to enter is CLASS. It happens to be positioned on line 4 of the screen. Press the Dup key. Notice CLASS appears under field name. If that is also to be our heading (as it is), simply press the Record Advance and we will proceed to our next item.

The next item to be entered is description. Press the roll up bar once. Notice description now appears on line 4 (DESC). Press the Dup key again. Instead of typing in DESC, it automatically appears through the depression of the Dup key. Either choice is acceptable. Whichever one is the most comfortable to you is the best one to use (i.e., whether to type in the variable name or to locate it and position it to the fourth line of your screen and press the Dup key).

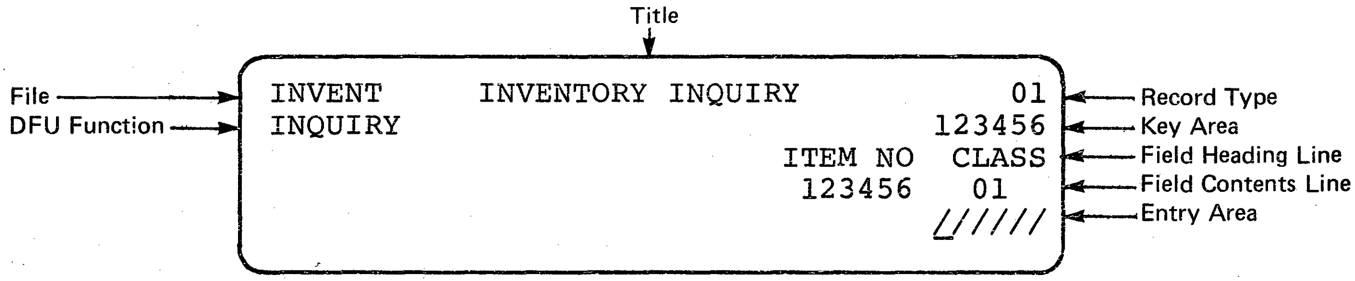
We want to enter a separate heading for description. Therefore, instead of pressing Record Advance, press Enter. The column heading prompt appears. At this point, type in the word DESCRIPTION and press the Enter key or Record Advance key as this is the last prompt for this field.

Now press roll up again. PRICE will appear. We are not interested in price for this inquiry. Press roll up again. ONHAND appears. We do want the on hand amount to be displayed during this inquiry. Press the Dup key and Enter since we want to have a different heading for our field. Then type in ON HAND with a space between the two words. Press the Enter key.

Again, roll up to ONORD, press the Dup key, Enter, type ON ORDER and Enter. Roll up to BKORD, press the Dup key, Enter and type BACK ORDER for our heading. Press the Enter key.

As there are no more field names that we need for our inquiries, press the Enter key without responding to the new field name prompt.

We now have the prompt asking to press the EOJ key or to update DFU specifications. We do not wish to update specifications in this exercise. We should go ahead therefore and print out the DFU specifications by pressing Command Key 6. After the specifications have printed, depress Command Key 7 to indicate the end of job. At that point, the message 'DFU DIAGNOSTICS ARE NOW IN PROGRESS' appears. If there are none, we will immediately begin executing the inquiry program. The resulting screen is identical to Figure 5.1. A description of what you see is also included in Figure 5.1.



```
INVENT       INVENTORY INQUIRY              01
INQUIRY                                 123456
                                ITEM NO  CLASS
                                 123456   01
                                        //////
```

Figure 5.1

Data from the first record automatically appears as we start our job. As we key in the item numbers for our inquiries, the screen will change to display the contents of those records.

At this point, we are ready to begin our inquiry. Look at the inventory file as shown in Figure 5.2.

INVENTORY MASTER FILE

REC CODE (1)	ITEM NO (2-7)	CLASS (8-9)	ITEM DESCRIPTION (10-21)	PRICE (22-26)	ON HAND (27-31)	ON ORDER (32-36)	BACK ORDER (37-41)	SALES M-T-D (42-46)	SALES Y-T-D (47-51)	DELETE CODE (52)
I	123456	1	WRENCH	8.00	100			100	300	
I	123457	4	NAILS	.50	400		50	25	300	
I	123458	3	BRUSH	2.50	25	500			50	
I	123459	2	NUTS	.50	400	1000		25	25	
I	123460	2	SCREWDRIVER	4.00	100			100	500	
I	123461	3	BOLTS	.75	400			10	10	
I	123462	1	PLIERS	4.25	100	1000		75	150	
I	123463	2	SCREWS	.30	200	500		100	600	
I	123464	1	HAMMER	3.00	50	500		150	500	
I	123465	1	SAW	10.00	40	500		25	550	

Figure 5.2

Let's inquiry into item number 123458. Key this now and press the Enter button. Immediately on the screen that item number appears along with its class - class number 3. To obtain its description which is the next field we specified, press Field Advance. The item description BRUSH appears. The next field in the record - PRICE - was not defined in our response to the prompts as a field we wished to display. Therefore, when we press Field Advance again, the next field defined - ONHAND - appears (with a balance of 25). Do this now.

Upon pressing Field Advance again, an on order amount of 500 is evident. A back order amount of 0 appears when it is pressed yet another time.

At this point, we could key in another item number that we may wish to inquire into. However, before doing that, we may wish to print out the data from our inquiry. We can do this by pressing Command Key 6. The headings will print along with the fields which we have just displayed. Press Command Key 6 now.

Let's do another inquiry - 123461. Enter this now. Field Advance twice to the on hand figure of 400. Let's say that this is the amount we wanted to see. We are not interested in the other two fields, on order and back order. But we do want to print out the results of this particular inquiry. Press Command Key 6. On order and back order also print. In this case, they happen to be 0. The printout always shows all the fields described regardless of what was displayed.

We may continue inquiring as often as we like in this manner until we are finished. Assuming this was our last inquiry, press Command Key 7 and then respond yes with Command Key 9 to the end of job prompt. This is the end of our inquiry run. Our listing should appear as shown in Figure 5.3.

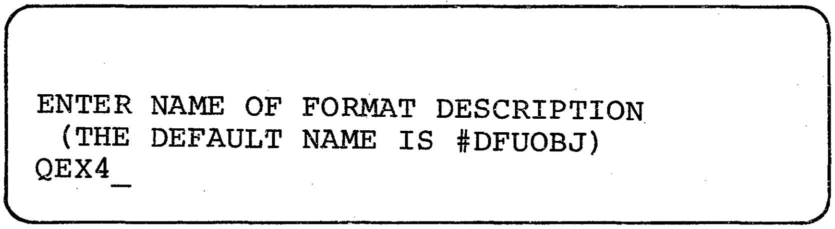
```
07/15/76              INVENTORY INQUIRY              PAGE 001

ITEM NO  CLASS  DESCRIPTION    ON HAND  ON ORDER  BACK ORDER

 123458   03    BRUSH           00025    00500       00000
 123461   03    BOLTS           00400    00000       00000
```

Figure 5.3

In the case of the INQUIRY and the UPDATE programs, you may wish to rerun them again and again as need for updates and inquiries appears. However, you do not have to re-enter the series of prompts everytime you wish to use the program. This is unnecessary. The first time the prompts are entered, they are stored after conversion by DFU into a module which we refer to as the format description. After the initial loading, this format description is referred to in order to rerun the job.

For example, we have just finished entering prompts for an inquiry job. We have also executed that inquiry job, completed it, and gone to an end of job. Now, let's say we wanted to do more inquiries of the same type; that is, into our item master file and displaying such things as class, description, and on hand, on order, and back order amounts. Our responses were all stored in the format description under the name QEX4. We can retrieve that now.

If you remember, the prompt asking for the format description included within parentheses 'THE DEFAULT NAME IS #DFUOBJ'. Take a look at Figure 5.4 to see that message.



```
ENTER NAME OF FORMAT DESCRIPTION
 (THE DEFAULT NAME IS #DFUOBJ)
QEX4_
```

Figure 5.4

If we had not keyed an entry (QEX4) at that point and simply pressed the Enter button, our resulting module would have been stored under the name #DFUOBJ. However, the next time a series of prompts was entered for another DFU job, the format description might have then become #DFUOBJ and the format description for our first job would have been replaced. The use of a unique name for the format description as we have been doing enables us to store the results of any one setup step (that is, responding to a series of prompts) by that unique name and making it recallable at any time in the future.

Let's recall the inquiry program now without having to respond to prompts. Re-enter the command INQUIRY. Having done that the next question asked is 'ENTER FILE NAME OF INQUIRY FILE'. Key INVENT. Next comes the prompt for format description. Key QEX4.

Since QEX4 is already stored as a result of our last run, upon pressing the Enter button after this entry has been made, all the attributes and specifications for that module are retrieved and the result is that our screen will display our inventory inquiry format immediately ready for execution. (As mentioned earlier, the first record in the file automatically appears at the beginning of execution.)

At this point, let's inquire into item number 123465. Key that number in now. Field Advance to display the fields. Press Command Key 6 to print if you desire. And then, since this is our only inquiry, press Command Key 7 for end of job. Then press Command Key 9 to answer the prompt - "END OF JOB?".

We have been able to execute our inquiry very quickly without going through the prompts. In other words, once the prompts have been answered, and that job needs to be run again, there is no need to re-enter them.

***

NOTE: The following example is for illustrative purposes only and should not be run on the System/32.

***

An even greater short cut may be taken by the use of the INQUIRY command statement. A command statement allows us to key what would be our responses to the first few prompts at the same time we enter the DFU function (INQUIRY). Instead of receiving the prompts to enter file name and the name of the format description, we could begin the inquiry job simply by typing the command statement as shown below and pressing the Enter key.

```
INQUIRY file name,format description name

INQUIRY INVENT,QEX4
```

This way, there are absolutely no prompts required at all, and our job would begin with a single command statement. This also may be used again anytime after the original prompts have been answered.

We will use command statements in later exercises.

THIS PAGE LEFT INTENTIONALLY BLANK

EXERCISE 5 - LIST FUNCTION

CLASS LISTING

Exercise 5 will be the first of three exercises to use the fourth and last function of DFU - the LIST function. In Exercise 5 we will be referencing the inventory master file - INVENT. Figure 6.1 is an illustration of what the report will look like.

```
07/15/76      GROUP LISTING       PAGE   1

CLASS    ON HAND  ON ORDER  BACK ORDER

  1        290      2000          0 *
  2        700      1500          0 *
  3        425       500          0 *
  4        400         0         50 *
          1815      4000         50 **
```

Figure 6.1

The listing will provide a grouping of on hand, on order, and back order balances by class for each of the four classes of inventory items (numbered 1 through 4). There will be no detail printing - only total printing for each class and then a final total for all classes reflecting total on hand, on order and back order quantities.

Let's enter the LIST command with its parameters rather than entering the parameters by responding to prompts. Type in the LIST command as illustrated below:

```
LIST file name,format description name,source member name

LIST INVENT,QEX5,QINVEN
```

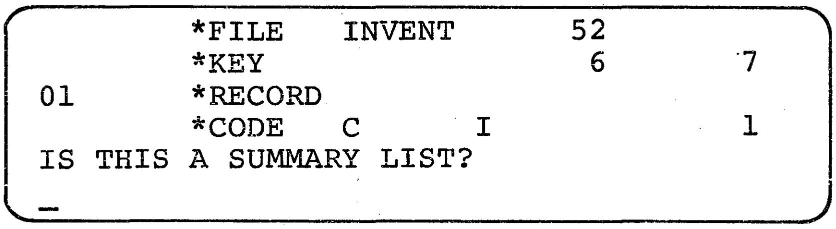
Notice the fourth entry (QINVEN - the source member name) is entered here and not in the inquiry command which we described in Exercise 4. This is because here the format description (QEX5) must yet be built and we must tell DFU that it will be using QINVEN. In Exercise 4, QEX4 was already built and as we had entered QINVEN in response to the source member name prompt originally, there was no need to repeat it in that INQUIRY command.

***

NOTE: The exact format of the LIST command in addition to the ENTER, UPDATE and INQUIRY commands is contained in the DFU Reference Manual, Appendix A (SC21-7600).

***

After the command has been keyed, press the Enter button. The message 'DFU ATTRIBUTES ARE BEING BUILT' will be displayed on the screen. Notice as we have entered the three parameters with the LIST command, the prompts for those parameters did not appear. As in the previous programs, the first four lines of the DFU attributes for QINVEN are displayed on the screen followed by the fifth line which is our first prompt. Look at Figure 6.2 to verify your display.



```
        *FILE   INVENT      52
        *KEY                 6       7
01      *RECORD
        *CODE   C       I            1
IS THIS A SUMMARY LIST?

_
```

Figure 6.2

The first question asks, "IS THIS A SUMMARY LIST?" - a summary list being those lists which include group totals. The answer is yes. Press Command Key 9.

The next prompt asks us on our summary list, "SHOULD DETAIL RECORDS BE PRINTED?". For this program we are only printing the class totals and detail records should not be printed. Press Command Key 10 to indicate no.

Column spacing value should be entered as 2 as previously. The title of our listing will be GROUP LISTING. Type this in now. The prompt "ANY FIELDS FROM THIS RECORD TYPE?" is again answered with a yes command from Command Key 9.

In the case of a listing, only three prompts are requested for each field - the field name, its heading and if the field is to be accumulated. There would be no prompt for auto-dup or for self-checking for a listing program.

DFU is now asking for the first field name to be used in our report. In this case it is CLASS. We can indicate CLASS simply by pressing the Dup button since it is the fourth line displayed on the screen. As the word CLASS is appropriate for our column title we do not need the field heading prompt to appear. Nor do we plan to accumulate CLASS. Therefore, press Record Advance after CLASS has been entered.

The next field name is requested. Use the roll up key and press it three times to arrive at ONHAND. Press the Dup button to record ONHAND as the next field which we will enter. This time press the Enter button after keying ONHAND to prompt 'ENTER COLUMN HEADING'. Enter the column heading as two separate words - ON HAND. Now press the Enter button because we want to be able to answer yes to the last prompt for the field - "ACCUMULATE THIS FIELD?". Now press Command Key 9 to respond yes.

Similarly we should now enter the on order information. Roll up the screen once, press Dup, press Enter, spell ON ORDER, press Enter and answer yes to "ACCUMULATE THE FIELD?".

Finally, the last field BKORD can be rolled up by pressing the roll up key one more time. Then press the Dup key followed by the Enter key, enter BACK ORDER, and press Enter to arrive at the prompt, "ACCUMULATE THIS FIELD?" to which we answer yes through Command Key 9.

The screen returns to another 'ENTER FIELD NAME' request. There are no more for this program. Simply press Enter. Now we will receive a series of new prompts unique to the LIST function of DFU.

The next question asked is "ARE THRE ANY CONTROL FIELDS?" Since this printout includes totals by class, CLASS would naturally be a control field. A control field is any field on which a change in that field will create a control break where printouts of totals for that group can occur.

In this case, we would want to print all accumulated totals for a particular control field - in this case CLASS. So, in response to the question if there are any control fields, answer yes through Command Key 9.

Next, we are asked for the name of that control field. In this case it is CLASS. Either roll down the screen five times or simply type the word CLASS followed by pressing the Enter button. DFU now asks if there is another control field. There could theoretically be many control fields. However, in this case there is only one - CLASS. Therefore a null response (that is, pressing the Enter button without any entry) will indicate that there are no more control fields.

The next question asks, "SHOULD THE FILE BE SORTED?". The answer to this is yes. If you will look at the inventory file listing in Figure 6.3 you will see that the items in the file are not in order by class.

INVENTORY MASTER FILE

REC	ITEM		ITEM		ON	ON	BACK	SALES	SALES	DELETE
CODE	NO	CLASS	DESCRIPTION	PRICE	HAND	ORDER	ORDER	M-T-D	Y-T-D	CODE
(1)	(2-7)	(8-9)	(10-21)	(22-26)	(27-31)	(32-36)	(37-41)	(42-46)	(47-51)	(52)
I	123456	1	WRENCH	8.00	100			100	300	
I	123457	4	NAILS	.50	400		50	25	300	
I	123458	3	BRUSH	2.50	25	500			50	
I	123459	2	NUTS	.50	400	1000		25	25	
I	123460	2	SCREWDRIVER	4.00	100			100	500	
I	123461	3	BOLTS	.75	400			10	10	
I	123462	1	PLIERS	4.25	100	1000		75	150	
I	123463	2	SCREWS	.30	200	500		100	600	
I	123464	1	HAMMER	3.00	50	500		150	500	
I	123465	1	SAW	10.00	40	500		25	550	

Figure 6.3

The first item number, 123456, is in class 1. The next is in class 4. The next is in class 3. Therefore, this file would be sorted before we would print it out. Use Command Key 9 to answer yes to this response.

Next, we are asked for the name of the sort field. In this case, since we want the records sorted by class we would indicate CLASS through the use of the roll-up, roll-down keys or through typing in the word CLASS. Indicate this now.

The next question asked about CLASS is, "IS IT ASCENDING?". In order to sort our file the System/32 has to know whether to sort the records into ascending or descending number by class. In this case ascending is correct. Press Command Key 9.

Next we are asked if there is another sort field named. Our response to this is that there is not and we would indicate this by simply pressing the Enter button without entering another name.

The next question asked is, "SELECT RECORDS BASED ON FIELD VALUES?". This is a function of DFU which will be used in Exercise 7 and explained at that time. For now the answer is no through Command Key 10.

That being the last prompt we now have the beginning of our DFU specifications displayed on the screen plus the request to hit the EOJ key (Command Key 7) or to update our DFU specifications. Press Command Key 6 (to print out our specifications) and then press the EOJ key. The message 'DFU DIAGNOSTICS ARE NOW BEING PERFORMED' is displayed after which we are about to begin execution of our program.

One final prompt is displayed, asking us whether the file is to be sorted or not. We indicated in our response to a prompt that the file is to be sorted. However, the presence of this prompt would also give us the ability not to sort the file by responding NOSORT to this prompt. In other words we have the ability to run the program each of two different ways any time we wish even though we have only loaded one format description.

In this case we wish to sort the file so type in the word SORT and press Enter. Upon doing that a message will be displayed indicating your data file is now being sorted. After the file is sorted, our listing will print out. At that time check it against the Figure 6.4 to ensure that all our entries were accurate.

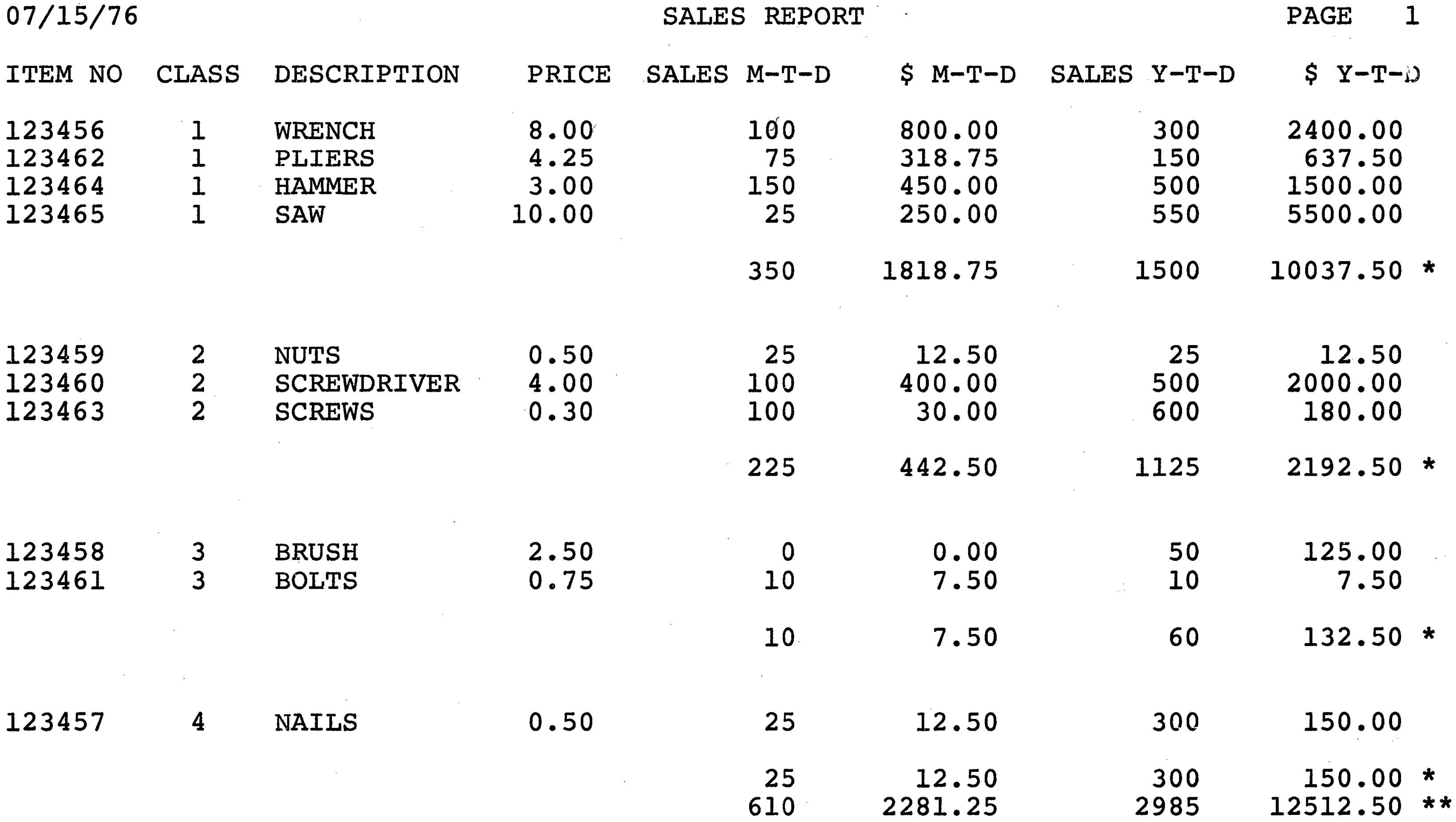
```
07/15/76      GROUP LISTING       PAGE   1

CLASS    ON HAND   ON ORDER   BACK ORDER

  1        290       2000          0 *
  2        700       1500          0 *
  3        425        500          0 *
  4        400          0         50 *
          1815       4000         50 *
```

Figure 6.4

EXERCISE 6 - LIST FUNCTION

SALES REPORT

Exercise 6 will again be a listing involving the inventory master file. However, as opposed to a summary listing this will be a detail summary listing. That is, group totals will print but also detail records will appear on the printout as shown in Figure 7.1.



```
07/15/76                               SALES REPORT                         PAGE   1

ITEM NO  CLASS  DESCRIPTION    PRICE  SALES M-T-D    $ M-T-D  SALES Y-T-D    $ Y-T-D

123456     1    WRENCH         8.00         100      800.00         300     2400.00
123462     1    PLIERS         4.25          75      318.75         150      637.50
123464     1    HAMMER         3.00         150      450.00         500     1500.00
123465     1    SAW           10.00          25      250.00         550     5500.00

                                            350     1818.75        1500    10037.50 *


123459     2    NUTS           0.50          25       12.50          25       12.50
123460     2    SCREWDRIVER    4.00         100      400.00         500     2000.00
123463     2    SCREWS         0.30         100       30.00         600      180.00

                                            225      442.50        1125     2192.50 *


123458     3    BRUSH          2.50           0        0.00          50      125.00
123461     3    BOLTS          0.75          10        7.50          10        7.50

                                             10        7.50          60      132.50 *


123457     4    NAILS          0.50          25       12.50         300      150.00

                                             25       12.50         300      150.00 *
                                            610     2281.25        2985    12512.50 **
```

Figure 7.1

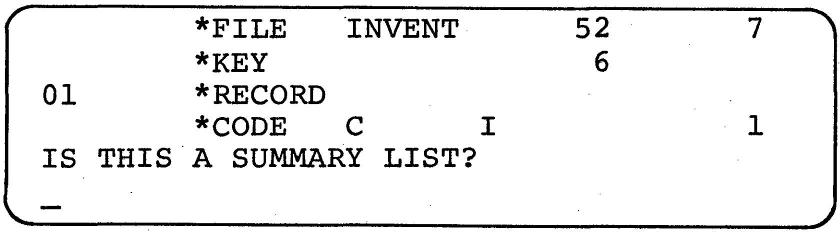
The file will be sorted into item number within class sequence as opposed to the strictly item number sequence in which it is now. In addition, we will multiply the sales month-to-date and year-to-date figures of each item by its price to arrive at a price of goods sold month-to-date and year-to-date. Therefore, on each detail line two extra fields will appear which are not in the record. Both will be derived by multiplying the price by one field or the other, sales month-to-date or sales year-to-date.

Control breaks will be taken on class and totals of the quantity month-to-date, the dollars month-to-date, the quantity year-to-date and the dollars year-to-date will be printed for each class. Totals will also print after the last record of the last class has been read to give us final totals for the report.

In addition, while loading our responses to prompts we will deliberately make an error so that you may see the error recovery procedure to be taken when prompts have been incorrectly responded to.

Now let us begin using the complete prompt method; that is, instead of using the entire LIST command we will simply type in the word LIST. This is followed by a prompt for the file name which is INVENT. This, in turn is followed by a request for the format description which we will enter as QEX6. The name of the source member is then requested (which is QINVEN) and this is entered.

At this point the DFU attributes are being built as is indicated by the display on the screen. The first four records in the DFU attributes for the source member QINVEN are then displayed along with the first prompt, "IS THIS A SUMMARY LIST?". This display is shown in Figure 7.2.



```
         *FILE   INVENT       52        7
         *KEY                  6
01       *RECORD
         *CODE   C      I               1
IS THIS A SUMMARY LIST?
_
```

Figure 7.2

A yes response to "IS THIS A SUMMARY LIST?" is required as we will be taking group totals. The next question is, "SHOULD DETAIL RECORDS BE LISTED?". A yes response should be indicated here. This is followed by the "SHOULD RECORD KEYS BE PRINTED?" prompt. In the case of INVENT, our inventory master file, the record keys are the item numbers and yes we do want the item numbers to print. Therefore answer yes to this question. The next question is, "ARE YOUR KEYS ALL NUMERIC?". Reply yes.

The column heading for keys should be ITEM NO. Column spacing is again 2. The title of the listing will be SALES REPORT. Enter this now and press the Enter key. Again the prompt "ANY FIELDS FROM THIS RECORD TYPE?" appears to which we answer yes.

Now we are asked for the first field name in this record type we will be using in our report. As item number was already described when we responded to the questions about keys, the first field we have to define is CLASS. Press the Dup key as CLASS is displayed on line 4 of our screen. Again the word CLASS is also appropriate for our heading and we do not want to accumulate it. Therefore press the Record Advance key.

The next field we wish to indicate is item description. Press the roll-up button once to get DESC onto the fourth line. Press the Dup key followed by the Enter key because we want the heading to be the actual word DESCRIPTION. Key this now and press the Record Advance key as we do not wish to (and cannot) accumulate this field.

Use the roll-up key once to arrive at the next field which will print which is PRICE. As in the case of CLASS (the field name), PRICE is appropriate for our heading and we do not want to accumulate the field. Therefore simply press the Dup key and Record Advance.

Now press the roll-up key four times to arrive at the sales month-to-date field (SLSMTD). This is the number of units we have sold for this particular item month-to-date. Press the Dup key to indicate that that will be our field name and then press the Enter button because we want to put in a different heading. Let's put in SALES M-T-D for sales month-to-date and then press the Enter button as we want to accumulate this field by answering yes to the next prompt. Now that we have that prompt, press Command Key 9 to indicate yes.

The next field we want to print out on our report is the dollar amount of the sales month-to-date. This field is not in the records of the file. Therefore, it must be created at this time.

To indicate that we are about to create a field not in a record press Command Key 12; that is, the CMD key followed by the plus-equal key to the right on the top line. This results in a prompt asking us to enter the length of the result field. As this field will be arrived at by multiplying price, a five position - two decimal field, by sales month-to-date - a five position - no decimal field, a result field of 10 is appropriate. Enter 10 now.

The next prompt asks for the number of decimal positions. As we are multiplying a 2 decimal field by a 0 decimal field the answer should be 2. Key this now.

The next prompt asks for the column heading. Key in $ M-T-D for dollars month-to-date.

The next question asked is should we accumulate this field. Use Command Key 9 to indicate yes.

Now we are asked for the first factor used in arriving at this result field. Use the Dup key to take sales month-to-date (SLSMTD) and place it on line 6. Then press the Enter key.

Now we are asked for the next factor in the result field. This is PRICE - the field that we wish to multiply sales month-to-date by. Either type this field or roll down to it and press the Dup button. Then press the Enter key.

The next prompt asks for the operation to be done - in this case multiply. MULT should be entered.

After we responded with MULT the next prompt asks for the next factor in the result field. This would be used in situations where the result would be arrived at through more than one operation which in this case was a multiply. For example, your result field might result from A plus B minus C divided by D. In this case you would need more than two factors to arrive at the result field. As there are no more factors to be entered in this calculation, leave the response line blank and press the Enter button.

The next prompt is 'ENTER FIELD NAME'. At this point we should enter our last field name from the record which is the sales amount year-to-date (SLSYTD). We would either roll up the screen to that field and press the Dup button or type in SLSYTD. However, to create an error later on type SLSYD instead of typing SLSYTD and press the Enter button.

The column heading is asked for next. Type SALES Y-T-D and press the Enter button. To the prompt "ACCUMULATE THIS FIELD?" enter yes.

One more field name must yet be defined. This would be the result of the price times the units year-to-date. Again, this is a field that we will be creating and which is not in our records. Therefore, again press Command Key 12 (the CMD key with the plus and equal key) to indicate that we are entering a new field.

Again enter 10 positions - 2 decimals, for the first two prompts. The column heading entered is $ Y-T-D. Press Enter. "ACCUMULATE THIS FIELD?". Press Command Key 9 to indicate yes.

Enter the first factor in result field (SLSYTD) and press Enter. Enter the next factor - PRICE. Key the operation MULT and press Enter.

The 'next factor' prompt receives a null response as there are no more factors to key at this point. The prompt then becomes "ENTER FIELD NAME". As there are no more field names either from the record or to be created, enter a null response to this prompt.

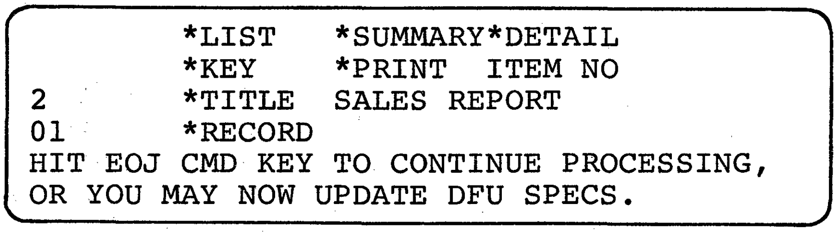
The next prompt asks if there are any control fields. As in the last exercise, indicate yes. This listing will include totals by class. Therefore one control field must be described - CLASS. Enter CLASS to the "ENTER CONTROL FIELD NAME" prompt. The next prompt is also "ENTER CONTROL FIELD NAME". If there were to be another control field it would be entered at this point. However we will use a null response to indicate that there will be none.

"SHOULD FILE BE SORTED?" - yes. Enter 'SORT FIELD NAME'. The file must be sorted into class in order to have our report print out by class. We also want to sort it into item number within each class so that these numbers will appear sequentially.

The major field CLASS should be keyed first. Do that now. The next question asks, "IS IT ASCENDING?". Press Command Key 9 for yes. Enter the next sort field - ITNUM. As it is also ascending, Command Key 9 is pressed to indicate yes to that prompt. We are asked for a third sort field. There are no more at this point. A null response should be taken. Press Enter.

The prompt, "RECORDS BASED ON FIELD VALUES?" is asked. Again this will be addressed in Exercise 7. For now respond no.

Now you should see the screen displaying the beginning of our specifications and asking us either to hit Command Key 7 or to update DFU specifications. It should resemble exactly Figure 7.3.



```
        *LIST   *SUMMARY*DETAIL
        *KEY    *PRINT  ITEM NO
2       *TITLE  SALES REPORT
01      *RECORD
HIT EOJ CMD KEY TO CONTINUE PROCESSING,
OR YOU MAY NOW UPDATE DFU SPECS.
```

Figure 7.3

Now let's print our DFU attributes and specifications by pressing Command Key 6. The printouts should be identical to Figures 7.4 and 7.5.

```
    ****  DFU ATTRIBUTES  ****

        *FILE   INVENT       52
        *KEY                  6        7
01      *RECORD
        *CODE   C      I               1
                ITNUM         6.0      7
                CLASS         2.0      9
                DESC         12       21
                PRICE         5.2     26
                ONHAND        5.0     31
                ONORD         5.0     36
                BKORD         5.0     41
                SLSMTD        5.0     46
                SLSYTD        5.0     51
                DCODE         1       52
```

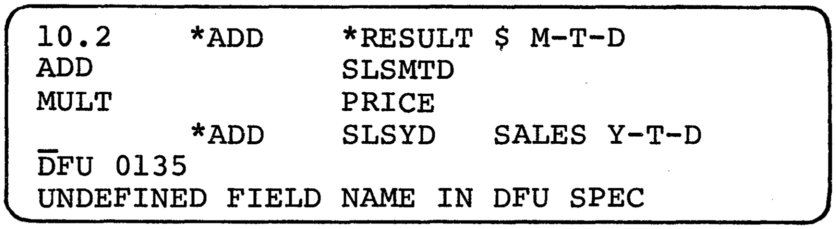
Figure 7.4

```
****  DFU SPECIFICATIONS  ****

        *LIST   *SUMMARY*DETAIL
        *KEY    *PRINT  ITEM NO
2       *TITLE  SALES REPORT
01      *RECORD
        *       CLASS   CLASS
        *       DESC    DESCRIPTION
        *       PRICE   PRICE
        *ADD    SLSMTD  SALES M-T-D
10.2    *ADD    *RESULT $ M-T-D
ADD             SLSMTD
MULT            PRICE
        *ADD    SLSYD   SALES Y-T-D
10.2    *ADD    *~~RESULT~~ $ Y-T-D
ADD             SLSYTD
MULT            PRICE
        *TOTAL  CLASS
        *SORTA  CLASS
        *SORTA  ITNUM
```

Figure 7.5

Now press Command Key 7 to indicate an end of job but remember that we have entered the field name for sales year-to-date incorrectly and that we will therefore have an error.

Notice the flashing screen. Press Error Reset to stop the flashing. The fourth line on the screen is always the line in error. This will always be the case as DFU goes through its diagnostics. DFU 0135 identifies the exact cause and included is a brief description of the problem, 'UNDEFINED FIELD NAME IN DFU SPECIFICATION'. Look at Figure 7.6.



```
10.2     *ADD     *RESULT $ M-T-D
ADD               SLSMTD
MULT              PRICE
_        *ADD     SLSYD   SALES Y-T-D
DFU 0135
UNDEFINED FIELD NAME IN DFU SPEC
```

Figure 7.6

It shows the screen as it is now. If necessary, a further definition of the error can be obtained by looking in the Messages Guide - Utilities Program Product (SC21-7618). and looking for DFU error #0135. However, in this case we can tell from the six word message appearing on line 6 exactly what the problem is. The field SLSYD should actually be SLSYTD. SLSYD was not used in QINVEN and is therefore not in our DFU attributes. We must now change it.

Notice the cursor is located at the first position of the line in error. To get to the field we want to change press Field Advance twice until we are at the incorrectly entered field SLSYD. Now type over it SLSYTD and press the Record Advance button as the heading SALES Y-T-D is correct. The message 'DFU DIAGNOSTICS ARE NOW IN PROGRESS' appears, indicating that DFU has accepted our change and is now continuing to look for yet other errors if there are any. There should be none and we should proceed with our program.

Again, complete information about reading and/or changing DFU specifications is contained in Chapter 10 of the DFU Reference Manual.

Again the prompt appears asking whether we should sort the file. In Exercise 7 we will show you both alternatives in use. However, here in Exercise 6 simply type SORT and press the Enter button. The message 'YOUR FILE IS NOW BEING SORTED' appears on the screen. When the file has finished being sorted our listing will appear. Notice the printout. Compare it to Figure 7.7.

```
07/15/76                               SALES REPORT                         PAGE   1

ITEM NO  CLASS  DESCRIPTION    PRICE  SALES M-T-D     $ M-T-D  SALES Y-T-D    $ Y-T-D

123456     1    WRENCH         8.00          100      800.00         300     2400.00
123462     1    PLIERS         4.25           75      318.75         150      637.50
123464     1    HAMMER         3.00          150      450.00         500     1500.00
123465     1    SAW           10.00           25      250.00         550     5500.00

                                             350     1818.75        1500    10037.50 *


123459     2    NUTS           0.50           25       12.50          25       12.50
123460     2    SCREWDRIVER    4.00          100      400.00         500     2000.00
123463     2    SCREWS         0.30          100       30.00         600      180.00

                                             225      442.50        1125     2192.50 *


123458     3    BRUSH          2.50            0        0.00          50      125.00
123461     3    BOLTS          0.75           10        7.50          10        7.50

                                              10        7.50          60      132.50 *


123457     4    NAILS          0.50           25       12.50         300      150.00

                                              25       12.50         300      150.00 *
                                             610     2281.25        2985    12512.50 **
```

Figure 7.7

Notice that all items from class 1 printed out first. Within class 1 the items are in order by item number. Similarly classes 2, 3 and 4 have printed out. For each class, totals have printed out for sales month-to-date, dollars month-to-date, sales year-to-date, and dollars year-to-date amounts in addition to final totals for those four fields. In addition, dollar month-to-date and dollar year-to-date figures have been calculated through multiplying price times sales month-to-date and sales year-to-date figures for each item.

EXERCISE 7 - LIST FUNCTION

RECEIPT LISTING

Exercise 7 will be a listing of item records from the transaction file. The entire transaction file is shown again in Figure 8.1 for reference purposes. It now includes those changes made during the update exercise (Exercise 3).

TRANSACTION FILE

REC	DFU	TRANS	TRANS	ITEM		DELETE
CODE	KEY	CODE	DATE	NO	QUAN	CODE
(1)	(2-6)	(7)	(8-13)	(14-19)	(20-24)	(25)
T	00010	I	081575	123456	40	
T	00020	I	080175	123458	10	
T	00030	A	080875	123458	50	
T	00040	R	080975	123458	500	
T	00050	A	081575	123459	100-	
T	00060	A	080875	123460	50	
T	00070	I	080875	123462	50	
T	00080	R	080175	123463	300	
T	00085	I	080275	123463	250	
T	00090	R	081275	123463	200	
T	00100	R	080675	123464	500	
T	00110	R	080175	123465	500	
T	00120	R	080375	123468	100	D

Figure 8.1

We will use two new facilities of DFU which have not been used in previous exercises. First, we will selectively list only receipts from our transaction file (those transactions indicated by an R in position 7). In addition, while we are printing the transaction file we would like the item lines that are being listed to include the item description. The item description is only in the item master file. We will chain or connect the transaction file records to their related master file records, take the description from the master file and include that in our printout of the transaction record. Our report will look like Figure 8.2.

```
07/15/76        RECEIPT REPORT        PAGE   1


CODE      DATE  ITEM.NO  DESCRIPTION       QUAN

  R     80175   123463   SCREWS            300
  R     80175   123465   SAW               500
  R     80675   123464   HAMMER            500
  R     80875   123458   BRUSH             500
  R     81275   123463   SCREWS            200

                                                     QUAN

                                                    2000 *
```

Figure 8.2

Now we will begin Exercise 7. Whereas before we had the option to enter a complete LIST command (or ENTER, UPDATE or INQUIRY command) with parameters or to respond to prompts which request the same information, in the case of a DFU program where we are chaining or connecting to a related master file, we must use the command with parameters.

In addition, during this exercise we will take advantage of one more facility of DFU - the storing on disk of our DFU attributes and specifications so that they may be modified later on to create an entirely different program without having to go through the entire prompting sequence. The ability to do this is also predicated on using a command with parameters to start the job.

The command we will enter is shown in Figure 8.3 (do not enter it now); that is, LIST TRANS (the transaction file we will be listing), QEX7 (the format description), QTRANS (the source member name) followed by the word SORT (or NOSORT) as we will be sorting our output (The entry of SORT within the command will eliminate the prompt - "SHOULD THE FILE BE SORTED BEFORE LISTING?" from appearing after the prompt sequence and immediately before execution.), followed by two commas, the letters NY, QEX7S, and INVENT.

```
LIST TRANS,QEX7,QTRANS,SORT,,NY,QEX7S,INVENT
```

Figure 8.3

***

NOTE: As each parameter has a specific position within the command (i.e., file name must be the first parameter, related master file the eighth, etc.), commas must be inserted to separate even unused (blank) parameters so that the correct positional relationship of the parameters can be maintained.

***

The NY that was entered relates to the saving of our DFU attributes and specifications. Entries at this point may have been NN, NY, YN, YY or GO. This will be the first exercise where we are saving our DFU attributes and specifications.

This is not to be confused with the entry of QEX7 as the second parameter. What that is is an unmodifiable module which resulted from entering our DFU specifications and gave us the ability to re-execute the program again without modification as we did during the inquiry exercise.

Here, however, we are also saving the DFU attributes and specifications. This gives us the added ability of modifying them later on if we wish.

Of the NY entry the first position is used to indicate if the source member for our DFU attributes and specifications already exists. The N indicates no. If the first character was a Y, the indication would be yes. The second character, which may also be a Y or an N, relates to what will happen with the resulting DFU specifications. If they are not to be saved, the second character would be an N. If they are to be saved, that character would be a Y.

As you can see, there are four possible combinations which may be entered here depending on our needs. The fifth possibility GO indicates the specifications are already stored and we do not wish to change them. It is similar to a YN response except that there will not be an opportunity to modify the already existing specifications even temporarily. A YN entry would allow you to modify the specifications but not store the changed specifications. If the changes must also be stored, a YY entry would be used here.

QEX7S is the name under which the DFU attributes and specifications resulting from our entries for Exercise 7 will be stored. If a module is already stored (YY or YN), the sixth parameter would be the name with which it is stored.

The last parameter, INVENT, appears because we are going to use a related master file to provide descriptive information for our transaction listing, in this case the item description. The last parameter after the LIST command is used to name this file.

Now key the LIST command as shown in Figure 8.3 and press the Enter button to begin building our attributes. The message 'DFU ATTRIBUTES ARE BEING BUILT' is displayed. This is followed by a new prompt, "KEY RPG II SOURCE NAME FOR ABOVE FILE". By keying in the LIST command with all its parameters we have eliminated the first three prompts that would normally appear (which request the file name, the format description name and the source member name). Now looking at your screen you will see above the request 'KEY RPG II SOURCE NAME FOR ABOVE FILE', the file name INVENT.

This prompt is activated by our entry of INVENT in the LIST command indicating that a related master file must be chained to. DFU is now asking for the name of the source member which identifies this file. It is QINVEN. Type that in now and press the Enter button.

The next prompt caused by the related master file parameter is "ENTER FIELD NAME FROM MASTER FILE KEY". The master file key is ITNUM for item number. This can be seen by referring to your listings of QINVEN. Enter ITNUM now.

After those entries DFU continues to build the attributes and then we once again arrive at the prompt - "IS THIS A SUMMARY LIST?". The answer to this is no as we only want a listing of the appropriate detail records and the final total. Press Command Key 10.

The next question is "SHOULD RECORD KEYS BE PRINTED?". This question should be answered no as the file we are referring to is the transaction file as can be seen on line 1 of the screen. If you remember the keys for these records are DFU generated keys and have no bearing on the printout of our report. Therefore use Command Key 10 to answer no.

Column spacing value is again 2. Enter RECEIPT REPORT for our title as this will be a listing of only our receipts from the transaction file. To the "ANY FIELDS FROM THIS RECORD TYPE?" prompt - answer yes. This is our transaction record and we will be printing from this record the following fields - the code which would be an R, the date, the item number, the description (taken from the master record) and the quantity.

The first field name for our report is CODE. It is already situated on line 4 of our screen. Therefore we need only press the Dup key to enter it on line 6. As the field name is appropriate for the heading and the field is obviously not to be accumulated we would press Record Advance to indicate the end of entries describing that field.

The next field is DATE. This can be obtained by pressing the roll-up key once to get DATE onto the fourth line, pressing the Dup key and again pressing Record Advance as no further entries are required.

The next field, item number, can be entered by again rolling up the screen one time to ITNUM and pressing the Dup key followed by the Enter key as we will want to give it a new column heading - in this case ITEM NO. After entering ITEM NO, press Record Advance as we do not wish to accumulate this field. Don't forget we are now describing the transaction file. Item number is not the key for TRANS and therefore had to be defined separately as we have just done.

The next field is from the master record - DESC. Type that in and press the Enter button. The column heading is asked for. At this point type in the word DESCRIPTION and press the Record Advance key as this field is not be accumulated.

***

<u>NOTE</u>: Fields will always appear on the printout in the order in which they were entered. Their order within a record (or within two different records as is the case here) does not effect the printout.

***

The next field we need is quantity. Type in QTY or use the roll-up key once and press the Dup key. Press the Enter button as we wish to enter a new heading - QUAN. Again press the Enter button as we wish to accumulate this field. Respond with Command Key 9 to the next prompt which asks us whether we want to accumulate the field.

As this is the last field needed to be described for our printout, enter a null response to the next "ENTER FIELD NAME" prompt. Enter no to the next prompt - "ANY CONTROL FIELDS?" as this will not be a summary report. We are merely listing the detail records and obtaining a final total.

The next prompt is "SHOULD THIS FILE BE SORTED?". Enter yes because we want to print out all the receipts in order by date. Press Command Key 9 and enter DATE when asked for the sort field name. Then depress Command Key 9 to indicate it is ascending and enter a null response to the next sort field name request.

***

NOTE: We can sort correctly on the six-position field DATE as the year is the same in each record (1975). If the year varied from record to record, it would have to be defined as a separate field in the source member (QTRANS). Then it could be entered as the first sort field in our responses in order to list the report in the proper sequence. Otherwise a record with a date of 080176 would print before one dated 081575.

***

Now we come to the prompt seen earlier in other exercises - "SELECT RECORDS BASED ON FIELD VALUES?". We only want to print those records that have an R in position 7; that is in the field CODE, as these are our receipts. Therefore, depress Command Key 9 to indicate - yes, we want to select records.

A new prompt appears asking us the name of the select field. In this case it is CODE. Press the Enter button after entering CODE. The next question asks for the condition. In this case we want a CODE which is equal to R. Equal is indicated by the EQ response. Type that in and press the Enter button.

The next question asks "IS FACTOR 2 A CONSTANT?". Yes, it is an R that will be constant throughout the running of the program. We will always be comparing the code to an R and looking for an equal condition. Answer yes.

***

NOTE: At another time, we might be comparing one field in a record to another. In this case, as both fields would be changing from one record to another, Factor 2 would not be a constant. In this exercise an R is being compared to a field in every record.

***

The next prompt asks for the entry of that constant. Type an R. Press the Enter button. Now we have finished describing our first select field. If there are no more select fields at this time, we would enter a null response to the prompt presently on the screen asking the next select field relation. However, in this exercise there is one other stipulation we wish to make. We do not want to print any records that have previously been deleted by a file update run. Any records with a D in position 25 should not be printed. Therefore, we have entered only the first condition.

Since the second condition is an AND relationship (that is, both the first and the second condition must be satisfied in order to select the record), we will type in AND. OR would have been entered if either the first or the second condition was sufficient to satisfy our requirement. Type in AND and press Enter.

The next prompt asks for the name of the select field. In this case it is DCODE. Type that in and press the Enter button. Our condition is not equal - NE. "IS FACTOR 2 A CONSTANT?". Press Command Key 9 indicating yes. Enter a D to the next prompt to indicate that this is the constant which DCODE must not be equal to in order to satisfy our requirements. After doing that, DFU asks if there is yet another select field relationship. In this case, indicate no by entering our response as a null response by pressing the Enter key.

***

NOTE: Instead of indicating a condition of DCODE as not equal to D, we could have responded by indicating DCODE should be equal to a blank. Either method would have sufficed.

***

At this point we have the option of pressing the End of Job key or updating DFU specifications. First, press Command Key 6 to print our DFU specifications. Having done that press Command Key 7 to end the job.

The message 'DFU DIAGNOSTICS ARE NOW IN PROGRESS' appears. There should be no diagnostics and the screen should soon state 'YOUR DATA FILE IS NOW BEING SORTED'. Notice that we are not asked at this point if the file is to be sorted. This was because we entered SORT in our command at the beginning of Exercise 7 when we typed LIST with its parameters. Therefore the data file is automatically being sorted now. The resulting printout should be identical to Figure 8.4.

```
07/15/76        RECEIPT REPORT        PAGE   1

CODE      DATE  ITEM NO  DESCRIPTION       QUAN

  R     80175   123463   SCREWS            300
  R     80175   123465   SAW               500
  R     80675   123464   HAMMER            500
  R     80875   123458   BRUSH             500
  R     81275   123463   SCREWS            200


                                                   QUAN

                                                  2000 *
```

Figure 8.4

Now let us show an example of the advantage of having stored the unmodifiable machine readable results of our DFU specifications as QEX7 rather than using #DFUOBJ. Re-enter the LIST command except this time we do not want the records sorted. Instead, we want the records to print out in item number sequence - the order in which they actually exist on the file.

The report we have just printed came from a temporary file into which the records were sorted and made available for writing out the report. This time we will print directly from the data file TRANS which is in order strictly by item number. The LIST command to do this is shown below.

```
        LIST TRANS,QEX7,,NOSORT,,,,INVENT
```

***

NOTE:

1. The source member name which holds the transaction file's RPG II specifications (QTRANS) does not have to be indicated here. It is only needed when you are creating the format description. As QEX7 was created by responding to the prompts before we ran the first receipt report, it is not necessary to specify QTRANS again.

2. As we are neither creating nor modifying stored DFU attributes and specifications during this run, there is no need to enter parameters for this. Therefore, neither parameters for creating or using a library member (NY, NN, YN, YY, GO) nor naming it (QEX7S) are required.

3. Notice that when parameters are removed from the command, all commas remain so that DFU knows the placement of all succeeding parameters.

***

Enter the LIST command now. Soon a message will appear on the screen that NOSORT is inconsistent with format. To stop the screen from flashing press the Error Reset key. Our format indicated that we would sort TRANS. This was indicated by our response yes to the prompt - "WILL THIS FILE BE SORTED?". The message we receive now is only a warning indicating that there is a discrepancy. An option of 0 should be taken to ignore the warning and continue with the job without sorting the file.

Again more information on any specific DFU error messages which you may encounter can be found in the System/32 Utilities Messages manual. Take a 0 option to the halt now and press the Enter button. Notice our report has just printed out again - this time in item number sequence as shown in Figure 8.5.

```
07/15/76        RECEIPT REPORT       PAGE   1

CODE      DATE  ITEM NO  DESCRIPTION      QUAN

  R    80875    123458   BRUSH            500
  R    80175    123463   SCREWS           300
  R    81275    123463   SCREWS           200
  R    80675    123464   HAMMER           500
  R    80175    123465   SAW              500

                                                  QUAN

                                                 2000 *
```

Figure 8.5

While the benefits of this flexibility are not so obvious with a file this small, remember that many files will be quite a bit larger and the sequence in which items are listed can make them more easily readable.

Now we will take advantage of the option available to store the DFU attributes and specifications for a particular job as we have done in this exercise. Let's say we required a similar listing except for issues which are indicated by an I in the code field or position 7 of your TRANS file. If we did not have the ability to store and modify DFU specifications we would have to go through the entire series of prompts again.

However, from our previous LIST command used when we ran the job the first time, the NY parameter indicated that we did not have DFU attributes and specifications stored at the beginning of the job (NY) but we did at the end (NY) and they are stored under the name QEX7S. Therefore, all that is needed is to re-enter the LIST command as before with the exception of changing the NY parameter to YN.

The first letter of the parameter (Y) now indicates that the DFU attributes and specifications are stored on the disk (under the name QEX7S). The N indicates that we do not want to store any changes that we will make at this time to that module. A Y here would change the program to print out only issues in the future. We want it to be permanently stored to print out only receipts and will change it only this time for issues.

Hence N is the second character in the parameter. Because a Y was entered as the first character of this parameter DFU will now give us an opportunity to change our specifications for this run only.

Another change we must make in addition to entering YN instead of NY is to change QEX7 to another name. QEX7 is already stored as a program which will print out receipts. By entering QEX8 instead (which does not presently exist), the system will know that we wish to create changes in the existing specifications (QEX7S) and store it in unmodifiable code separately as QEX8. Again the entry of QEX8 as the second parameter will only cause the object code which is unalterable by the operator to be stored. Thereby the 'ISSUE' program could be run again but the specifications which we will be creating for it will not be stored (a YY entry would store it but eliminate the old entry) and further modifications to create other programs would be made to the specifications for the 'RECEIPT' program.

Therefore change the second parameter QEX7 to QEX8 which does not as of now exist. Notice that with a single set of stored attributes and specifications we can easily create many different programs by modifying that code.

We will leave the NOSORT parameter as it is. Now enter the LIST command as shown below.

```
    LIST TRANS,QEX8,QTRANS,NOSORT,,YN,QEX7S,INVENT
```

The first prompt after the entry of the command asks the source member name of the related file to which we again key in QINVEN. If we were not using a related master file no prompts would appear at all. Again the next prompt asks for the name of the record key - to which we again key in ITNUM. After doing that, all subsequent prompts are skipped and we go right to that point where we may press the End of Job key or update the specifications.

At this point we wish to update the specifications. As the first four lines of our specifications are displayed and we want to change the constant R to the constant I press the roll-up key until we come to that line. Refer to Figure 8.6 for guidance.

```
****  DFU SPECIFICATIONS  ****

         *LIST   *RECORD *DETAIL
         *KEY
2        *TITLE  RECEIPT REPORT
01       *RECORD
         *       CODE    CODE
         *       DATE    DATE
         *       ITNUM   ITEM NO
         *       DESC    DESCRIPTION
         *ADD    QTY     QUAN
         *SORTA  DATE
         *SELECT CODE    EQ
                 R
AND      *SELECT DCODE   NE
                 D
```

Figure 8.6

Eight taps of the roll-up button should bring you to the line where there is only the letter R. This immediately follows the line which states SELECT CODE EQ so we know that this is the line we want to change.

These two lines taken in conjunction state SELECT those records where the field CODE is EQ (equal) to an R.

Press Field Advance twice to position the cursor underneath the R. Type an I, press Field Advance again and then Record Advance to record our change.

Don't forget one other change we have to make. The title of the report should be changed from RECEIPT REPORT to ISSUE REPORT. To do this, press the roll-down button until we come to the title line (the third line of our specifications). Press Field Advance twice to position the cursor under the title RECEIPT REPORT. Then type ISSUE REPORT directly over it remembering to blank out the final R and T which have not been overlapped because of the shorter entry. Then, press Field Advance and Record Advance to record the change.

As there are no more changes to be done we may now press the EOJ key, Command Key 7, to indicate that we are at an end of job and that we are to begin execution. Again the message 'DFU DIAGNOSTICS ARE NOW IN PROGRESS', appears and this is again followed by the warning that the entry NOSORT on our command is inconsistent with the format. Press the Error Reset key, key a zero to ignore the warning and the report will begin printing the unsorted records.

All our issues have now printed out in item number sequence as shown in Figure 8.7 as we specified NOSORT.

```
07/15/76          ISSUE REPORT          PAGE   1

CODE      DATE   ITEM NO  DESCRIPTION      QUAN

  I     81575    123456   WRENCH            40
  I     80175    123458   BRUSH             10
  I     80875    123462   PLIERS            50
  I     80275    123463   SCREWS           250
                                                     QUAN

                                                     350 *
```

Figure 8.7

You have now completed the last exercise in the course and have been introduced to all of the facilities of DFU. Of course, the brevity of the course does not allow introduction to every possibility that might happen while you are running DFU. Its intention is to make you feel comfortable with the exercises herein and also leave you able to reference the appropriate manuals when necessary in order to obtain additional information.

You should clear the System/32 of the information you stored in it during this course. If you were able to use the names of the data files and exercise members as printed in the course the following commands will clear the system of all practice information. If, because your company had already used any of the suggested names you had to use names of your choice, simply substitute the names you used in these commands to clear the system.

```
DELETE  INVENT,F1    REMOVE  QEX1,LOAD     REMOVE  QEX5,LOAD
DELETE  TRANS,F1     REMOVE  QEX2,LOAD     REMOVE  QEX6,LOAD
REMOVE  QINVEN       REMOVE  QEX3,LOAD     REMOVE  QEX7,LOAD
REMOVE  QTRANS       REMOVE  QEX4,LOAD     REMOVE  QEX8,LOAD
REMOVE  QEX7S
```

READER'S COMMENT FORM

SR30-0069-1

IBM System/32
Data File Utility (DFU)
Facilities and Operations
Student Text

Please comment on the usefulness and readability of this publication, suggest additions and deletions, and list specific errors and omissions (give page numbers). All comments and suggestions become the property of IBM. If you wish a reply, be sure to include your name and address.

COMMENTS

- Thank you for your cooperation. No postage necessary if mailed in the U.S.A.
  FOLD ON TWO LINES, STAPLE AND MAIL

Your comments, please . . . . .

Your comments on the other side of this form will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Fold Fold

First Class
Permit 40
Armonk
New York

Business Reply Mail

No postage stamp necessary if mailed in the U.S.A.

Postage will be paid by:

International Business Machines Corporation
Education Development, Dept. 71G
P. O. Box 2150
Atlanta, Georgia 30301

Fold Fold

IBM®

International Business Machines Corporation

General Systems Division
4111 Northside Parkway N.W.
P.O. Box 2150
Atlanta, Georgia 30301
(U.S.A. only)

General Business Group/International
44 South Broadway
White Plains, New York 10601
U.S.A.
(International)

IBM ®

International Business Machines Corporation

General Systems Division
4111 Northside Parkway, N.W.
P.O. Box 2150
Atlanta, Georgia 30301
(U.S.A. only)

General Business Group/International
44 South Broadway
White Plains, New York 10601
U.S.A.
(International)