

## HP 3000 RPG/VIEW INTERACTIVE PROCESSING TOPICS

Christopher J. Colburn

This paper is geared towards the beginning RPG programmer on the HP3000 and is intended to demonstrate how interactive data handling programs can be created on the HP3000 by using the RPG/View interface. This short paper includes a sample screen program in Appendix A which is presently in use at the G.D. Armstrong Company. The author wishes to state that the program may or may not be the best written RPG program available and any comments or suggestions by the readers of this article would be greatly appreciated.

The author also assumes that the readers of this article have a good knowledge of both RPG and VIEW/3000.

### INTRODUCTION

Have you ever purchased a product from a department store, gotten it home, and become frustrated because the product was harder to use than it was first anticipated? If so, and if you are an RPG programmer working on the HP3000 series of computers, then this paper is dedicated especially to you!

Even though documentation in the RPG Manual on the RPG/View interface is very thorough when dealing with batch programming, the discussion of how to apply interface to interactive programming is very lax. The example provided for programmers in the manual is the RPG version of VIEW/3000's "ENTRY" program, but this does not show an entry-level RPG programmer how to write a program which will search files, update existing records, display existing fields of a record, and add records to a file. Since documentation on creating interactive screens on the HP3000 in

RPG is sparse, and because I feel it is unfair to put anyone else through the misery of searching through manuals and testing many programs, I have decided to write this paper on using VIEW/3000 and RPG on the HP3000 series of computers. I have also included an example in Appendix A to help the first time RPG programmer utilize the interface for many programming needs.

When designing a good computer system, the programmer/analyst must try to make it user friendly. Since an operator/user will have most of the day-to-day interaction with the hardware and software on the system, it is imperative that the entry of data be designed in such a way as to encourage productivity. One way to do this is to make the user completely comfortable entering data into the system.

There are several ways to enter data on the HP3000. Data can be entered through the utilities of the programming language used, through the V/3000 ENTRY program, or interactively using a combination of programming language utilities and V/3000.

Programming languages like COBOL and RPG have commands which allow the user to enter data into the system by the programming language. For example, COBOL uses the "ACCEPT" command to accept data from the user at the terminal. RPG uses the "DSPLY" command to enter data through the system. These commands are good to use when entering in one or two fields of information, or if a small number of people are using a particular system, but are limited and slow when handling large amounts of information about a record or when a large number of people are using the system.

The "ENTRY" program provided by Hewlett Packard for use with V/3000 is

adequate and easy to use, but has major faults. On one hand, it provides for easy maintenance of screens without the writing of bulky programs. Editing of screen fields is also handled by VIEW. All items are keyed into a batch file, reformatted, and then placed into a KSAM, IMAGE, or MPE file. ENTRY, however, has overriding negative factors to discourage its use. Some of these factors are that ENTRY does not handle data interactively (on-line processing), ENTRY only updates records within the batch file, and ENTRY does not determine whether the record you are working on is in the main file until after reformatting.

The best kind of data entry system is one which will provide maximum data with minimum effort on the part of the user. This type of information system can be achieved on the HP3000 by using a combination of one of the various programming languages (FORTRAN, COBOL, BASIC, RPG, etc) and V/3000. FORTRAN, COBOL, and BASIC allow the programmer to code in calls to V/3000 intrinsics which build the VIEW screens. RPG, because of its structure on the HP3000, has to internally call these intrinsics. Thus, it is not as easy to call V/3000 intrinsics for the RPG programmer as it is for programmers in other languages. Status words are not available to the RPG programmer, and calling V/3000 screens in RPG seems almost impossible.

Each language, however, has its tools to create screens on the HP3000, and RPG is no exception. Because chapter 13 of the RPG MANUAL (RPG INTERFACE TO V/3000) is documented in detail, I will only summarize the major functions of the RPG-V/3000 interface in this paper. These functions are illustrated in the sample program displayed in Appendix A.

## GETTING STARTED

I started writing RPG interactive screen programs by playing with the batch RPG screen program, ENTRYRPG. I compiled that program, created a screen to enter test data, and keyed in some data. The program worked okay, but I wanted (and my company needed) something better. My company and I needed a dynamic data handling screen program which would call up a record in a file, check to see if the chosen record was on file to update it, or enter the new record. In short, we needed a program which would make full use of the capabilities of the machine we were using. Because the RPG/VIEW interface seemed only able to handle batch records, our company even looked at going with a second language such as COBOL or BASIC just to write our screen programs. The monthly maintenance was too expensive to run just these screen programs, and a total conversion from RPG to COBOL, or

RPG to BASIC also was too time-consuming and costly for us. After examining the ENTRYRPG program, I realized that it was geared totally to the "batch" method of entering data. Since my company needed an interactive data handling screen program, and I had no examples from which to create one, I "robbed" the best parts of ENTRYRPG and started to create an interactive data handling screen using RPG.

As in COBOL handling of screens, "top down" programming in the RPG handling of screens is next to impossible. I did, however, try to "standardize" as much of the actions and events that I possibly could. For example, any action which is called from the program has been placed into a standard subroutine. Then, all I had to do is call that subroutine (through the "EXSR" command), and the action was executed.

I also found out that the editing of data (negative numbers, right justify, blanking fields, etc) was handled very poorly by the RPG/VIEW interface. If a negative number was entered into a numeric field, the program would accept the negative number. However, if you were going to look at this field when updating the record, the interface will overlay the negative sign over the last digit on the right. V/3000 will not accept the number as numeric. Because of this I decided to define numeric fields which could be both positive or negative as character fields. I added 2 characters to each field (one for a sign, one for a decimal point), and created my own edit subroutines on all fields going to or coming from a screen.

## CREATING THE INTERACTIVE SCREEN

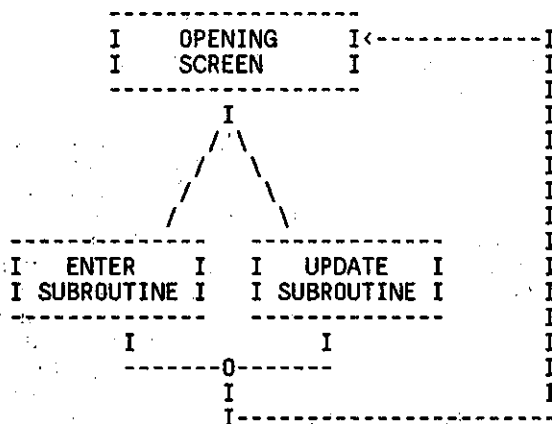
First, as a point of review:

The RPG/VIEW interface does not strictly follow the RPG logic cycle. All indicators must be set on or off manually before any action is performed. The EXCPT, READ, and CHAIN commands dominate this type of program. Subroutines to edit fields going to or coming from the screen are needed, for the RPG/VIEW interface handles negative numbers very poorly. This interface also uses what are called "ACTIONS" and "EVENTS". ACTIONS are output records written to the WORKSTN file. EVENTS are input records read from the WORKSTN file. More information about ACTIONS, EVENTS, the RPG/VIEW interface is available in Chapter 13 of the HP3000 RPG Manual.

One of the first things to do in the design of a screen is to try to picture what the screen is going to do. In the example provided (Appendix A), a screen was needed to look at an Accounts Receivable entry file. The file was a KSAM file, so I knew that I would need a first

or "front" screen with the key field on it. Once the key was entered on this screen, I knew that the record had to be added to the file (ENTER subroutine) or had to be updated (UPDATE subroutine). After the record was added or up-

dated, the program was to return to the "front" screen and await either entry of another key or termination of the program. I pictured it like this:



Now that I had an idea of what course my program was to follow, I wrote it, following these basic steps:

1. Define the input/update files in the "F" specs.
2. Define the single WORKSTN file (TRANSFILE) on the "F" specs.
3. Set up edit field and message arrays in the "E" specs.
4. Define the input/update file fields on the "I" specs.
5. Define the screen fields on the "I" specs. Multiple screens are like a file that has different types of records; each type of record is defined on the input specs. Similarly, each different screen is defined on the input specs.
6. Define WORKSTN function keys.
7. Bring first screen up and decide whether the record is to be entered or updated (refer to Appendix A).

a. Move form name to form name field.

b. Set up the correct repeat and freeze form options.

c. Move form name field to FORMSPEC NEXTFORM option (EXSR INTFRM).

d. Specify next form name and repeat/freeze options (EXSR CHGSUB).

e. Set the next form from forms file, set repeat/freeze options, and initialize fields according to FORMSPEC (EXSR INTSUB).

f. Reset the form - clear the fields (EXSR GETSUB).

g. Replace data in V/3000 data buffer with data from user program buffer (EXSR PUTSUB).

h. Display current form, initial data, and messages. Read input from terminal to V/3000 data buffer and read WORKSTN record (EXSR SHWSUB).

i. After the screen is read, clear the message window (EXSR CLRMSG).

- j. Check to see if either a function key or the ENTER key was hit. If an invalid function key was hit, a message will notify the operator. If ENTER was hit, perform edits specified for the fields in the current form. This is followed by a read of the WORKSTN file to check for errors (EXSR EDTSUB). If errors are found, go back to "h" above and try again.
- k. If the screen passed the edits, then write the data from the data data buffer to the user program buffer, then do a read of the WORKSTN file (EXSR GETSUB).
- l. Move any important screen data which you do not want to lose to a hold field or area.
- m. Move user program buffer to the data buffer (EXSR PUTSUB).
- n. Perform a chain on KSAM files to see if you are going to enter or update the record. Then enter or update the record.
- o. Set up the first screen again and set off all indicators. Then go back to the beginning of the cycle and start all over again. If you wish to end the program, function key "F8" has been coded to end the program.

8. When entering or updating a record, you follow a similar pattern to the one just described in step 7. When updating, however, you must first move the fields from the record being updated to the screen and then perform the actions previously described.

9. Also: In the sample program in Appendix A, note that the WORKSTN file (TRANSFIL) has a continuation ("K") record. Unless you define your VIEW form as "FORMA", you will need an MPE file equation to run this program. This equation would be similar to:

:FILE FORMA=GDAR4FM.PUB.GD

Finally, to summarize this paper, I wish to state that effective interactive screen programs on the HP3000 can be created with the RPG/VIEW programming interface. Because, in my opinion, the documentation in the RPG Manual did not properly address interactive on-line programming, I felt that this paper was necessary. At the very minimum, a working example has been provided with this paper, and it is the author's hope that RPG programmers can benefit from his labor on this topic. Hopefully, this information will lead towards standardized, well written, and easily maintainable RPG programs on the HP3000.

\*\*\*\*\* APPENDIX A \*\*\*\*\*

HDUMPFIL

```

F**-----**
F** THIS SCREEN PROGRAM UTILIZES SUBROUTINES TO CALL **
F** ACCOUNTS RECEIVABLE SCREENS. FORMA GDAR4FM ENTERS, **
F** UPDATES, OR DELETES A/R GDENTER RECORDS. **
F** CHRISTOPHER J. COLBURN, GD ARMSTRONG COMPANY 082583. **
F**-----**
F** DEFINE INPUT AND SCREEN FILE LENGTHS **
F**-----**
FGDENTER UC F 128 128R08AI 3 DISC A
FTRANSFILUD V 299 WORKSTN L 5
F KFORMS FORMA

```

```

E*-----*
E* SET UP PROGRAM EDIT FIELD AND MESSAGE ARRAYS *
E*-----*
E      CARA      9 1
E      ARA       7 1
E      ARA1      7 1
E      ARA2      6 1
E      MSG1      1 1 60
E      MSG2      1 1 60
E      MSG3      1 1 60
E      MSG4      1 1 60
E      MSG5      1 1 60
E      MSG6      1 1 60
E      MSG7      1 1 60
E      MSG8      1 1 60
E      MSG9      1 1 60
E      MSG10     1 1 60
E      MSG11     1 1 60
E      MSG12     1 1 60
E      MSG13     1 1 60
E      MSG14     1 1 60

```

```

I*-----*
I* DEFINE INPUT/UPDATE FILE FIELDS *
I*-----*
IGDENTER AA 01
I      1 2 GERCDE
I      3 10 GREFN
I      11 160GEDATE
I      17 24 GEITEM
I      25 49 GEDESC
I      50 550GECUST
I      56 80 GENAME
I      P 81 823GETAXP
I      83 830GECOD
I      84 840GETAXC
I      85 850GEPAID
I      86 860GECPOST
I      87 870GEIPST
I      88 92 BLANK1
I      93 930GEINVC
I      94 950GEDLOC
I      96 970GE0C
I      98 121 BLANK2
I      P 98 1001GEQTY
I      P 101 1034GEPRIC
I      P 104 1064GEUCST

```

74

```

C*-----*
C* MOVE FORM NAME TO FORM NAME FIELD *
C*-----*
C          BEGIN      TAG
C          MOVE "GDAR7"  FORMB
C          MOVE "0"      RPTAPP 1
C          MOVE "0"      FRZAPP 1
C          EXSR INTFRM
C*-----*
C* SPECIFY NEXT FORM NAME AND REPEAT/APPEND OPTIONS *
C*-----*
C          EXSR CHGSUB
C          START      TAG
C*-----*
C* GET THE NEXT FORM FOR USE AND INITIALIZE THE FIELDS*
C* NOTICE THAT 91 IS SET ON. THIS IS USED WHEN CALLING*
C* ACTION "PUTDTA", AND DETERMINES WHICH SCREEN IS *
C* BEING PROCESSED. *
C*-----*
C          SETON          91
C          EXSR INTSUB
C*-----*
C* DO GETDTA HERE TO CLEAR/RESET SCREEN *
C*-----*
C          EXSR GETSUB
C*-----*
C* MOVE LAST KEY PROCESSED TO FRONT SCREEN TO TRACK *
C* WHERE OPERATOR IS. *
C*-----*
C          MOVE BFREFN  SAREFN
C*-----*
C* REPLACE DATA IN V/3000 DATA BUFFER WITH DATA IN *
C* USER PROGRAM BUFFER. (MOVE DATA FROM BUFFER TO SCN *
C*-----*
C          EXSR PUTSUB
C* 3 * DISPLAY FORM
C          REPEAT      TAG
C          SETOF          111213
C          SETOF          141516
C          SETOF          171920
C          SETOF          747522
C          SETOF          767778
C          SETOF          7966
C*-----*
C*SHOW FORM AND PLACE INPUT IN DATA BUFFER. *
C*-----*
C          EXSR SHWSUB
C*-----*
C* 6 * CLEAR MESSAGE WINDOW *
C*-----*
C          EXSR CLRMSG
C*-----*
C* SEE IF FUNCTION KEY OR ENTER KEY IS HIT HERE. IF F2*
C* OR F8 NOT HIT, THE KEY IS INVALID AND A MSG IS SENT*
C* TO THE OPERATOR. IF ENTER, CONTINUE. *
C*-----*
C          78          SETON          LR
C          LR          GOTO EXIT          EXIT IF F8
C          75          SETON          76
C          77          SETON          76
C* INVALID KEY (F1-F7) HIT - TRY AGAIN
C          76          MOVE MSG1      MSG      60
C          76          EXSR MSGD02
C          76          GOTO REPEAT
C*-----*
C* IF ENTER HIT - RUN VIEW EDITS *

```

```

C*-----*
C          EXSR EDTSUB
C*** IF ERRORS - RETURN TO REPEAT AND TRY AGAIN
C 74          GOTO REPEAT
C*-----*
C* TRANS DATA FROM DATA BUF TO UP BUFF, THEN READ REC*
C*-----*
C          EXSR GETSUB
C* PERFORM USER EDITS
C          MORERR TAG
C*-----*
C* MOVE 1ST SCN INFO INTO HOLD AREA FOR DSPLY THE NEXT*
C* TIME THE FORM IS SHOWN.
C*-----*
C          MOVE SAREFN BFREFN 8
C          EXSR PUTSUB
C          PCHAIN TAG
C*-----*
C* CHAIN KEY ON SCREEN TO SEE IF YOU ARE ENTERING OR *
C* UPDATING THE RECORD. 22 ON IF REC NOT ON FILE.
C*-----*
C          BFREFN CHAINGDENTER 22
C          SETOF 91
C          SETON 92
C N22          EXSR UPDATE
C 22          EXSR ENTER
C*-----*
C* RESET BACK TO FIRST FORM AND REDO
C*-----*
C          MOVE "GDAR7" FORMB
C          MOVE "0" RPTAPP
C          MOVE "0" FRZAPP
C          EXSR INTFRM
C*-----*
C* SPECIFY NEXT FORM NAME AND REPEAT/APPEND OPTIONS *
C*-----*
C          EXSR CHGSUB
C*-----*
C* RETURN TO START (AFTER HOUSECLEANING)
C*-----*
C          SETOF 111213
C          SETOF 141516
C          SETOF 171920
C          SETOF 226791
C          SETOF 937475
C          SETOF 767778
C          SETOF 796692
C          SETOF 94
C          GOTO START
C* END OF PROGRAM/PROCESSING
C          EXIT TAG
C*****
C***** ENTER SUBROUTINE *****
C*****
CSR          ENTER BEGSR
C*-----*
C* MOVE 2ND FORM NAME TO FORM NAME FIELD
C*-----*
CSR          ENTST TAG
CSR 92          MOVE "GDAR8" FORMB
CSR          MOVE "0" RPTAPP
CSR          MOVE "0" FRZAPP
CSR          EXSR INTFRM
C*-----*
C* SPECIFY NEXT FORM NAME AND REPEAT/APPEND OPTIONS *
C*-----*

```



```

CSR          EXSR CHGSUB
CSR          ESTART    TAG
C*-----*
C* GET THE NEXT FORM FOR USE AND INITIALIZE THE FIELDS*
C* NOTICE THAT 92 IS SET ON. THIS IS USED WHEN CALLING*
C* ACTION "PUTDTA", AND DETERMINES WHICH SCREEN IS *
C* BEING PROCESSED. *
C*-----*
CSR          EXSR INTSUB
C*-----*
C* DO GETDTA HERE TO CLEAR/RESET SCREEN *
C*-----*
CSR          EXSR GETSUB
C*-----*
C* MOVE 1ST SCREEN INFO TO 2ND SCREEN *
C*-----*
CSR 92          MOVE BFREFN    SCREFN

C*-----*
C* REPLACE DATA IN V/3000 DATA BUFFER WITH DATA IN *
C* USER PROGRAM BUFFER. (MOVE DATA FROM BUFFER TO SCN *
C*-----*
CSR          EXSR PUTSUB
C* IF MASTER REC ADDED, DISPLAY MESSAGE
CSR          REPT    TAG
CSR          SETOF          111213
CSR          SETOF          141516
CSR          SETOF          171920
CSR          SETOF          7475
CSR          SETOF          767778
CSR          SETOF          79
C*-----*
C*SHOW FORM AND PLACE INPUT IN DATA BUFFER. *
C*-----*
CSR          EXSR SHWSUB
C*-----*
C* 6 * CLEAR MESSAGE WINDOW *
C*-----*
CSR          EXSR CLRMSG
C*-----*
C* SEE IF FUNCTION KEY OR ENTER KEY IS HIT HERE. IF F2*
C* OR F8 NOT HIT, THE KEY IS INVALID AND A MSG IS SENT*
C* TO THE OPERATOR. IF ENTER, CONTINUE. *
C*-----*
CSR 78          GOTO ENDENT
CSR 77          SETON          76
CSR 75          SETON          76
CSR 76          MOVE MSG1      MSG
CSR 76          EXSR MSGD02
CSR 76          GOTO REPT
C*-----*
C* IF ENTER HIT - RUN VIEW EDITS *
C*-----*
CSR          EXSR EDTSUB
C*** IF ERRORS - RETURN TO STEP 3
CSR 74          GOTO REPT
C*** IF NO ERRORS, CONTINUE
C*-----*
C* TRANS DATA FROM DATA BUF TO UP BUFF, THEN READ REC*
C*-----*
CSR          EXSR GETSUB
C*-----*
C* MOVE 2ND SCN INFO INTO HOLD AREA FOR DSPLY THE NEXT*
C* TIME THE FORM IS SHOWN. *
C*-----*
CSR 92          EXSR HLDBF1

```

```

C*-----*
C* REPLACE DATA IN V/3000 DATA BUFFER WITH DATA IN *
C* USER PROGRAM BUFFER. *
C*-----*
C          EXSR PUTSUB
C*-----*
C* ADD DETAIL LINE OF INVOICE HERE *
C*-----*
CSR          SETON          81
CSR          EXCPT
CSR          SETOF          81
C*-----*
C* EXIT SUBROUTINE (AFTER HOUSECLEANING) *
C*-----*
CSR          SETOF          111213
CSR          SETOF          141516
CSR          SETOF          171920
CSR          SETOF          747576
CSR          SETOF          777879
C*-----*
C* ROUTINE TO SWAP FORMS *
C*-----*
CSR          SETON          91
CSR          SETOF          92
C*-----*
C* END OF PROCESSING *
C*-----*
CSR          ENDENT      TAG
CSR          ENDSR
C*-----*

C*****
C*****      UPDATE SUBROUTINE      *****
C*****
CSR          UPDATE      BEGSR
C*-----*
C* MOVE FORM NAME TO FORM NAME FIELD *
C*-----*
CSR          UPDST      TAG
CSR 92          MOVE "GDAR8"      FORMB
CSR          MOVE "0"      RPTAPP
CSR          MOVE "0"      FRZAPP
CSR          EXSR INTERM
C*-----*
C* SPECIFY NEXT FORM NAME AND REPEAT/APPEND OPTIONS *
C*-----*
CSR          EXSR CHGSUB
C*-----*
C* GET THE NEXT FORM FOR USE AND INITIALIZE THE FIELDS*
C* NOTICE THAT 92 IS SET ON. THIS IS USED WHEN CALLING*
C* ACTION "PUTDTA", AND DETERMINES WHICH SCREEN IS *
C* BEING PROCESSED. *
C*-----*
CSR          EXSR INTSUB
C*-----*
C* DO GETDTA HERE TO CLEAR/RESET SCREEN *
C*-----*
CSR          EXSR GETSUB
C*-----*
C* GET INFO FROM DATA BUF TO USER PROG BUF (SCREEN) *
C*-----*
CSR 92          EXSR MVSCN1
C*-----*
C* REPLACE DATA IN V/3000 DATA BUFFER WITH DATA IN *
C* USER PROGRAM BUFFER. (MOVE DATA FROM BUFFER TO SCN *
C*-----*
CSR          EXSR PUTSUB
C* 6US * DISPLAY FORM

```

```

CSR          REPD          TAG
CSR          SETOF          111213
CSR          SETOF          141516
CSR          SETOF          171920
CSR          SETOF          677475
CSR          SETOF          767778
CSR          SETOF          7966
C*-----*
C*SHOW FORM AND PLACE INPUT IN DATA BUFFER.      *
C*-----*
CSR          EXSR SHWSUB

C*-----*
C* 6 * CLEAR MESSAGE WINDOW                        *
C*-----*
CSR          EXSR CLRMSG
C*-----*
C* SEE IF FUNCTION KEY OR ENTER KEY IS HIT HERE. IF F2*
C* OR F8 NOT HIT, THE KEY IS INVALID AND A MSG IS SENT*
C* TO THE OPERATOR. IF ENTER, CONTINUE.            *
C*-----*
CSR 78          GOTO EXITSR
CSR 75          SETON          66
CSR 77          SETON          76
CSR 76          MOVE MSG1      MSG
CSR 76          EXSR MSGD02
CSR 76          GOTO REPD
C*-----*
C* IF ENTER HIT - RUN VIEW EDITS                    *
C*-----*
CSR          EXSR EDTSUB
C*** IF ERRORS - RETURN TO STEP 3
CSR 74          GOTO REPD
C*-----*
C* TRANS DATA FROM DATA BUF TO UP BUFF, THEN READ REC*
C*-----*
CSR          EXSR GETSUB
C*-----*
C* MOVE 2ND SCN INFO INTO HOLD AREA FOR DSPLY THE NEXT*
C* TIME THE FORM IS SHOWN.                          *
C*-----*
CSR 92          EXSR HLDBG1
C*** SUPPLY USER EDITS HERE
C*-----*
C* REPLACE DATA IN V/3000 DATA BUFFER WITH DATA IN *
C* USER PROGRAM BUFFER.                             *
C*-----*
CSR          EXSR PUTSUB
C*-----*
C* UPDATE RECORD WITH INFO FROM SCREEN TO FILE        *
C*-----*
CSR          SETON          82
CSR          EXCPT
CSR          SETOF          82
C*-----*
C* EXIT SUBROUTINE (AFTER HOUSECLEANING)              *
C*-----*
CSR          SETOF          111213
CSR          SETOF          141516
CSR          SETOF          171920
CSR          SETOF          677475
CSR          SETOF          767778
CSR          SETOF          797173
CSR          SETOF          66
C*-----*
C* ROUTINE TO SWAP FORMS                            *

```

```

C*-----*
CSR          SETON          91
CSR          SETOF          92
C*-----*
C* END OF PROCESSING      *
C*-----*
CSR          EXITSR      TAG
CSR          ENDSR
C*****
CSR          HLDBF1      BEGSR
CSR          MOVE SCRCOD      BFCOD  2
CSR          MOVE SCREFN      BFREFN  8
CSR          Z-ADDSCDATE      BFDATE  60
CSR          MOVE SCITEM      BFITEM  8
CSR          MOVE SCDESC      BFDESC  25
CSR          Z-ADDSCCUST      BFCUST  60
CSR          MOVE SCNAME      BFNAME  25
CSR          Z-ADDSCSTAX      BFTAX   33
CSR          Z-ADDSCCOD      BFCOD   10
CSR          Z-ADDSCSTAXC     BFTAXC  10
CSR          Z-ADDSCPAID      BFPAID  10
CSR          Z-ADDSCCPST      BFCPST  10
CSR          Z-ADDSCIPST      BFIPST  10
CSR          MOVE SCBLNK      BFBLNK  5
CSR          Z-ADDSCINVC      BFINVC  10
CSR          Z-ADDSCDLOC      BFDLOC  20
CSR          Z-ADDSCGEOC      BFGEOC  20
C* EDIT SCREEN TO FILE FIELDS
CSR          MOVEASCQTY      CARA
CSR          EXSR NMEDIT
CSR          EDTFLD      MULT 10      QTYFLD  51
CSR          Z-ADDQTYFLD      BFQTY   51
C* EDIT PRICE (SCREEN TO FILE) FIELD
CSR          MOVEASCPRIC      CARA
CSR          EXSR NMEDIT
CSR          EDTFLD      MULT .01      PRCFLD  54
CSR          Z-ADDPRCFLD      BFPRIC  54
C* EDIT COST (SCREEN TO FILE) FIELD
CSR          MOVEASCCOST      CARA
CSR          EXSR NMEDIT
CSR          EDTFLD      MULT .01      PRCFLD
CSR          Z-ADDPRCFLD      BFCOST  54
C* EDIT PRICE PER GALLON FIELD
CSR          MOVEASCPRCG      CARA
CSR          EXSR NMEDIT
CSR          EDTFLD      MULT .01      PRCFLD
CSR          Z-ADDPRCFLD      BFPRPG  54

C* MOVE MD AND FED PERCENTAGE TO EDIT BUFFER
CSR          Z-ADDSCMPCT      BFMPCT  33
CSR          Z-ADDSCFPCT      BFFPCT  33
C* EDIT EXTENSION FIELD
CSR          MOVEASCEXTN      CARA
CSR          EXSR NMEDIT
CSR          Z-ADDEDTFLD      BFEXTN  72
C* EDIT ROADTAX FIELD
CSR          MOVEASCRTAX      CARA
CSR          EXSR NMEDIT
CSR          Z-ADDEDTFLD      BFRTAX  72
C* EDIT SALETAX FIELD
CSR          MOVEASCSTAX      CARA
CSR          EXSR NMEDIT
CSR          Z-ADDEDTFLD      BFSTAX  52
C* EDIT TOTAL FIELD
CSR          MOVEASCTOTL      CARA
CSR          EXSR NMEDIT
CSR          Z-ADDEDTFLD      BFTOTL  72

```

```

CSR                                ENDSR
C*-----*
CSR          HLDBG1  BEGSR
CSR          MOVE SCRCOD  BGRCOD  2
CSR          MOVE SCREFN  BGREFN  8
CSR          Z-ADDSCDATE  BGDATE  60
CSR          MOVE SCITEM  BGITEM  8
CSR          MOVE SCDESC  BGDESC  25
CSR          Z-ADDSCCUST  BGCUST  60
CSR          MOVE SCNAME  BGNAME  25
CSR          Z-ADDSTAX  BGTAX  33
CSR          Z-ADDSCCOD  BGCOD  10
CSR          Z-ADDSTAXC  BGTAXC  10
CSR          Z-ADDSCPAID  BGPAID  10
CSR          Z-ADDSCCPST  BGCPST  10
CSR          Z-ADDSCIPST  BGIPST  10
CSR          MOVE SCBLNK  BGBLNK  5
CSR          Z-ADDSCINVC  BGINVC  10
CSR          Z-ADDSCDLOC  BGDLOC  20
CSR          Z-ADDSCGEOC  BGGEOC  20
C* EDIT SCREEN TO FILE FIELDS
CSR          MOVEASCQTY  CARA
CSR          EXSR NMEDIT
CSR          EDTFLD  MULT 10  QTYFLD  51
CSR          Z-ADDQTYFLD  BGQTY  51
C* EDIT PRICE (SCREEN TO FILE) FIELD
CSR          MOVEASCPRIC  CARA
CSR          EXSR NMEDIT
CSR          EDTFLD  MULT .01  PRCFLD  54
CSR          Z-ADDPFCFLD  BGPRIC  54

C* EDIT COST (SCREEN TO FILE) FIELD
CSR          MOVEASCCOST  CARA
CSR          EXSR NMEDIT
CSR          EDTFLD  MULT .01  PRCFLD
CSR          Z-ADDPFCFLD  BGCOST  54
C* EDIT PRICE PER GALLON FIELD
CSR          MOVEASCPRCG  CARA
CSR          EXSR NMEDIT
CSR          EDTFLD  MULT .01  PRCFLD
CSR          Z-ADDPFCFLD  BGPRPG  54
C* MOVE MD AND FED PERCENTAGE TO EDIT BUFFER
CSR          Z-ADDSCMPCT  BGMPCT  33
CSR          Z-ADDSCFPCT  BGFPCT  33
C* EDIT EXTENSION FIELD
CSR          MOVEASCEXTN  CARA
CSR          EXSR NMEDIT
CSR          Z-ADDEDTFLD  BGEXTN  72
C* EDIT ROADTAX FIELD
CSR          MOVEASCRTAX  CARA
CSR          EXSR NMEDIT
CSR          Z-ADDEDTFLD  BGRTAX  72
C* EDIT SALETAX FIELD
CSR          MOVEASCSTAX  CARA
CSR          EXSR NMEDIT
CSR          Z-ADDEDTFLD  BGSTAX  52
C* EDIT TOTAL FIELD
CSR          MOVEASCTOTL  CARA
CSR          EXSR NMEDIT
CSR          Z-ADDEDTFLD  BGTOTL  72
CSR          ENDSR
C*-----*
CSR          MVSCNI  BEGSR
CSR          MOVE GERCDE  SCRCOD
CSR          MOVE GREFN  SCREFN
CSR          Z-ADDGDATE  SCDATE
CSR          MOVE GEITEM  SCITEM

```

CSR		MOVE GEDESC	SCDESC
CSR		Z-ADDGECUST	SCCUST
CSR		MOVE GENAME	SCNAME
CSR		Z-ADDGETAXP	SCTAX
CSR		Z-ADDGECOD	SCCOD
CSR		Z-ADDGETAXC	SCTAXC
CSR		Z-ADDGEPAID	SCPAID
CSR		Z-ADDGECPT	SCCPT
CSR		Z-ADDGEIPST	SCIPST
CSR		MOVE BLANK1	SCBLNK
CSR		Z-ADDGEINVC	SCINVC
CSR		Z-ADDGEDLOC	SCDLOC
CSR		Z-ADDGEOC	SCGEOC

C\* EDIT QTY FILE TO SCREEN FIELD

CSR		SETON		70
CSR	GEQTY	MULT .1	FLDQTY	72
CSR		Z-ADDFLDQTY	TESTFD	72
CSR		EXSR EDITSB		
CSR		MOVE EARA	SCQTY	

C\* EDIT PRICE (FILE TO SCREEN) FIELD

CSR		SETON		69
CSR	GEPRIC	MULT 100	FLDPRC	52
CSR		Z-ADDFLDPRC	TESTFD	
CSR		EXSR EDITSB		
CSR		MOVE EARA	SCPRIC	

C\* EDIT COST (FILE TO SCREEN) FIELD

CSR		SETON		69
CSR	GEUCST	MULT 100	FLDPRC	
CSR		Z-ADDFLDPRC	TESTFD	
CSR		EXSR EDITSB		
CSR		MOVE EARA	SCCOST	

C\* EDIT PRICE PER GALLON (FILE TO SCREEN) FIELD

CSR		SETON		69
CSR	GEPRPG	MULT 100	FLDPRC	
CSR		Z-ADDFLDPRC	TESTFD	
CSR		EXSR EDITSB		
CSR		MOVE EARA	SCPRCG	

C\* MOVE MD AND FED PERCENTAGE TO EDIT BUFFER

CSR		Z-ADDGEMPCT	SCMPCT	
CSR		Z-ADDGEFPCT	SCFPCT	

C\* EDIT EXTENSION FIELD

CSR		Z-ADDGEEXTN	TESTFD	
CSR		EXSR EDITSB		
CSR		MOVE DARA	SCEXTN	

C\* EDIT ROADTAX (FILE TO SCREEN) FIELD

CSR		Z-ADDGERTAX	TESTFD	
CSR		EXSR EDITSB		
CSR		MOVE DARA	SCRTAX	

C\* EDIT SALETAX FIELD

CSR		SETON		68
CSR		Z-ADDGESTAX	TESTFD	
CSR		EXSR EDITSB		
CSR		MOVE EARA	SCSTAX	

C\* EDIT TOTAL (FILE TO SCREEN) FIELD

CSR		Z-ADDGETOTL	TESTFD	
CSR		EXSR EDITSB		
CSR		MOVE DARA	SCTOTL	
CSR		ENDSR		

C\*\*\*\*\*  
C\*\*\*\*\* CLEAR MESSAGE WINDOW \*\*\*\*\*  
C\*\*\*\*\*  
CSR CLRMSG BEGSR  
CSR MOVE "60" MSLEN 2  
CSR MOVE "J" ENHCD  
CSR MOVE MSG2 MSG

```

CSR          SETON          50
CSR          EXCPT
CSR          SETOF          50
CSR          MOVE "SHOW " ACTION
CSR          SETON          55
CSR          EXCPT
CSR          SETOF          55
CSR          ENDSR
C*****
C* THIS SUBROUTINE EDITS NUMBERS FROM FILE TO SCREEN
CSR          EDITSB        BEGSR
CSR          MOVE MSG2     CARA
CSR          MOVE MSG2     ARA
CSR          MOVE MSG2     ARA1
CSR          MOVE MSG2     ARA2
CSR          TESTFD        COMP 0          61
CSR 61        TESTFD        MULT 2          TESTF1 72
CSR 61        TESTFD        SUB TESTF1     TSTFD1 72
CSR          Z-ADD TESTFD   TSTFD1
CSR          MOVE TSTFD1    TEST1 7
CSR          MOVE ATEST1    ARA
CSR 61        MOVE "-"      CARA,1
CSR          MOVE " "      CARA,1
CSR          Z-ADD0        X
CSR          LOOP01        TAG
CSR          X             ADD 1          X
CSR          ARA,X         COMP "0"      60
CSR 60        MOVE " "      ARA,X
CSR          X             COMP 7        62
CSR 62        GOTO ENDL01
CSR 60        GOTO LOOP01
CSR          ENDL01        TAG
CSR          Z-ADD0        X
CSR          Z-ADD1        Y
C* ARA,1 THRU ARA,5 MOVED TO CARA,2 THRU CARA,6.
CSR          LOOP02        TAG
CSR          X             ADD 1          X
CSR          Y             ADD 1          Y
CSR          MOVE ARA,X     CARA,Y
CSR          X             COMP 5        63
CSR          GOTO LOOP02
CSR          MOVE "."      CARA,7
CSR          MOVE ARA,6     CARA,8
CSR          MOVE ARA,7     CARA,9

C* GO TO 5 OR 6 DIGIT EDITS OR EXIT THIS SUBROUTINE
CSR 70        GOTO SUBR1
CSR 69        GOTO SUBR2
CSR 68        GOTO SUBR3
CSR 67        GOTO SUBR4
CSR          GOTO ENDSR1
C* ROUTINE TO CONVERT 5.1 FIELDS
CSR          SUBR1        TAG
CSR          MOVE CARA,9    ARA1,7
CSR          MOVE " "      ARA1,6
CSR          MOVE CARA,8    ARA1,5
CSR          MOVE CARA,6    ARA1,4
CSR          MOVE CARA,5    ARA1,3
CSR          MOVE CARA,4    ARA1,2
CSR          MOVE CARA,1    ARA1,1
CSR          GOTO ENDSR1
C* ROUTINE TO CONVERT 5.4 FIELDS
CSR          SUBR2        TAG
CSR          MOVE CARA,9    ARA1,7
CSR          MOVE CARA,8    ARA1,6
CSR          MOVE CARA,6    ARA1,5
CSR          MOVE CARA,5    ARA1,4

```

```

CSR      MOVE " "      ARA1,3
CSR      MOVE CARA,4    ARA1,2
CSR      MOVE CARA,1    ARA1,1
CSR      GOTO ENDSR1
C* ROUTINE TO CONVERT 5.2 FIELDS
CSR      SUBR3          TAG
CSR      MOVE CARA,9    ARA1,7
CSR      MOVE CARA,8    ARA1,6
CSR      MOVE CARA,7    ARA1,5
CSR      MOVE CARA,6    ARA1,4
CSR      MOVE CARA,5    ARA1,3
CSR      MOVE CARA,4    ARA1,2
CSR      MOVE CARA,1    ARA1,1
CSR      GOTO ENDSR1
C* ROUTINE TO CONVERT 6.0 FIELDS
CSR      SUBR4          TAG
CSR      MOVE CARA,9    ARA2,6
CSR      MOVE CARA,8    ARA2,5
CSR      MOVE CARA,6    ARA2,4
CSR      MOVE CARA,5    ARA2,3
CSR      MOVE CARA,4    ARA2,2
CSR      MOVE CARA,3    ARA2,1
C* END OF EDITS
CSR      ENDSR1        TAG
CSR      MOVEACARA      DARA      9
CSR      MOVEAARA1      EARA      7
CSR      MOVEAARA2      FARA      6
CSR      MOVEAMSG2      ARA
CSR      MOVEAMSG2      CARA
CSR      MOVEAMSG2      ARA1
CSR      MOVEAMSG2      ARA2
CSR      SETOF          606162
CSR      SETOF          636768
CSR      SETOF          6970
CSR      ENDSR

```

C\*\*\*\*\*

C\* THIS SUBROUTINE EDITS NUMBERS FROM SCREEN TO FILE..

```

CSR      NMEDIT        BEGSR
CSR      MOVEAMSG2      ARA
CSR      MOVEAMSG2      ARA1
CSR      MOVEAMSG2      ARA2
CSR      Z-ADD1          X      20
CSR      Z-ADD1          Y      20
CSR      CARA,1          COMP "-"      60
CSR 60      MOVE " "      CARA,1
CSR      Z-ADD9          X
CSR      LOOP3          TAG
CSR      X              COMP 0      62
CSR 62      GOTO ENDL03
CSR      CARA,X          COMP " "      61
CSR 61      X              SUB 1      X
CSR 61      GOTO LOOP3
CSR      ENDL03          TAG
C* IF FIELD IS BLANK (X=0), RESET INDEX X TO 9
CSR      X              COMP 0      65
CSR 65      Z-ADD9          X
CSR      Z-ADD7          Y
CSR      LOOP4          TAG
CSR      X              COMP 0      63
CSR      Y              COMP 0      63
CSR 63      GOTO ENDL04
CSR      CARA,X          COMP " "      64
CSR 64      X              SUB 1      X
CSR 64      GOTO LOOP4
CSR      MOVE CARA,X      ARA,Y
CSR      X              SUB 1      X

```



```

CSR      Y      SUB 1      Y
CSR      GOTO LOOP4
CSR      ENDL04      TAG
CSR      MOVEAARA      TEST02 7
CSR      MOVE TEST02      TEST03 70
C* QTY EDITED (70 ON), NO DECIMALS, SO Z-ADD TO EDITI
CSR      TEST03      MULT .01      EDITI 72
CSR 60      EDITI      MULT -1      EDTFLD 72
CSR N60      EDITI      MULT 1      EDTFLD
CSR      MOVEAMSG2      CARA
CSR      MOVEAMSG2      ARA
CSR      SETOF      606162
CSR      SETOF      636465
CSR      ENDSR
C*-----*
C*V/3000 SCREEN SUBROUTINES
C*-----*
CSR      INTERM      BEGSR
CSR      MOVE "      "NXTFRM 15
CSR      MOVE" "      "NXTFRM
CSR      MOVEFORMB      NXTFRM
CSR      ENDSR
C*-----*
C*SUBROUTINE TO CHANGE FORMS-B
CSR      CHGSUB      BEGSR
CSR      SETON      58      CHGNXT-50
CSR      EXCPT
CSR      SETOF      58
CSR      ENDSR
C*-----*
C*SUBROUTINE TO GET NEXT FORM AND INITIALIZE-C
CSR      INTSUB      BEGSR
CSR      MOVE"GETNXT" ACTION 6
CSR      SETON      55      GETNXT-51
CSR      EXCPT
CSR      SETOF      55
C* SET INITIAL VALUES
CSR      MOVE"INIT " ACTION
CSR      SETON      55      INIT-58
CSR      EXCPT
CSR      SETOF      55
CSR      ENDSR
C*-----*
C*SHOW THE FORM AND PLACE INPUT IN V/3000 DATA BUF-D
CSR      SHWSUB      BEGSR
CSR      MOVE"SHOW " ACTION
CSR      SETON      55      SHOW-53
CSR      EXCPT
CSR      SETOF      55
C* READ FROM TERMINAL
CSR      MOVE"RDTERM" ACTION
CSR      SETON      55      RDTERM-54
CSR      EXCPT
CSR      SETOF      55
CSR      READ TRANSFIL      H0
CSR      ENDSR
C*-----*
C*SUBROUTINE TO PERFORM USER EDITS IF ENTER HIT-E
CSR      EDTSUB      BEGSR
CSR      MOVE"EDITS " ACTION
CSR      SETON      55      EDITS-59
CSR      EXCPT
CSR      SETOF      55
CSR      READ TRANSFIL      H0
CSR      ENDSR
C*-----*

```

```

C*SUBROUTINE TO GET DATA BUF INFO TO USER PROG BUF-F
CSR      GETSUB      BEGSR
CSR      MOVE "GETDTA" ACTION      55      GETDTA-64
CSR      SETON
CSR      EXCPT
CSR      SETOF      55
CSR      READ TRANSFIL      H0
CSR      ENDSR
C*-----*
C*SUBROUTINE TO PUT USER PROG BUF INFO TO DATA BUF-G
CSR      PUTSUB      BEGSR
CSR      SETON      57      PUTDTA-63
CSR      EXCPT
CSR      SETOF      57
CSR      ENDSR
C*-----*
C*SUBROUTINE TO PUT USER PROG BUF INFO TO DATA BUF-G
CSR      SHDSUB      BEGSR
CSR      SETON      59      SHODTA-57
CSR      EXCPT
CSR      SETOF      59
CSR      ENDSR
C*-----*
CSR      MSGD01      BEGSR
CSR      MOVE "60"      MSLEN
CSR      MOVE "J"      ENHCD
CSR      MOVE MSG1      MSG
CSR      SETON      50
CSR      EXCPT
CSR      SETOF      50
CSR      ENDSR
C*-----*
CSR      MSGD02      BEGSR
CSR      MOVE "60"      MSLEN 2
CSR      MOVE "J"      ENHCD 1
CSR      SETON      50
CSR      EXCPT
CSR      SETOF      50
CSR      ENDSR
C*-----*
CSR      MSGD03      BEGSR
CSR      MOVE "60"      MSLEN 2
CSR      MOVE "J"      ENHCD 1
CSR      SETON      51
CSR      EXCPT
CSR      SETOF      51
CSR      ENDSR
C*-----*
CSR      MSGD04      BEGSR
CSR      MOVE "60"      MSLEN 2
CSR      MOVE "J"      ENHCD 1
CSR      SETON      52
CSR      EXCPT
CSR      SETOF      52
CSR      ENDSR
C*-----*

0*** INDICATOR 55 FOR ACTIONS 51,53,54, 58-61, 64-70
0*** INDICATOR 56 FOR ACTIONS 56 AND 62
0*** INDICATOR 57 FOR ACTIONS 57 AND 63
0*** INDICATOR 58 FOR ACTION 50
0TRANSFILE      55
0      ACTION      6
0      E      56
0      6 "CORERR"
0      FLDNO      11

```

```

0      MSLEN      13
0      ENHCD      14
0      MSG        75
0      E          50
0
0      MSLEN      6 "PUTMSG"
0      ENHCD      8
0      MSG        9
0      E          52
0
0      MSLEN      6 "SHOMSG"
0      ENHCD      8
0      MSG        9
0      E          51
0
0      FLDNO      6 "BADFLD"
0      MSLEN      11
0      ENHCD      13
0      MSG        14
0* THIS IS THE PUTDTA FOR ENTRY DATA.....
0      E          57 91
0
0      DATA LN   6 "PUTDTA"
0      SAREFN     10
0      E          57 92
0
0      DATA LN   6 "PUTDTA"
0      SCRCOD     10
0      SCREFN     12
0      SCDATE     20
0      SCITEM     26
0      SCDESC     34
0      SCCUST     59
0      SCNAME     65
0      SCTAX      90
0      SCCOD      93
0      SCTAXC     94
0      SCPAID     95
0      SCCPST     96
0      SCIPST     97
0      SCBLNK     98
0      SCINVC     103
0      SCDLOC     104
0      SCGEOC     106
0      SCQTY      108
0      SCPRIC     115
0      SCCOST     122
0      SCPRCG     129
0      SOMPCT     136
0
0      SCFPCT     139
0      SCEXTN     142
0      SCRTAX     151
0      SCSTAX     160
0      SCTOTL     167
0      E          58
0
0      NXTFRM     6 "CHGNXT"
0      RPTAPP     21
0      FRZAPP     22
0* SETON 82 ONLY IF IT IS TO UPDATE FILE; IF NOT SETOF 82
0      OGDENTER EADD 81
0      BFRCD      2
0      BREFN      10
0      BFDATE     16
0      BFITEM     24
0      BFDESC     49

```

0			BFCUST	55
0			BFNAME	80
0			BFTAX	82P
0			BFCOD	83
0			BFTAXC	84
0			BFPAID	85
0			BFCPST	86
0			BFIPST	87
0			BFBLNK	92
0			BFINVC	93
0			BFDL0C	95
0			BFGEOC	97
0			BFQTY	100P
0			BFPRIC	103P
0			BFCOST	106P
0			BFPRPG	109P
0			BFMPCT	111P
0			BFFPCT	113P
0			BFEXTN	117P
0			BFRTAX	121P
0			BFSTAX	124P
0			BFTOTL	128P
0	EDEL	82 66		
0			BGRCOD	2
0			BGREFN	10
0			BGDATE	16
0			BGITEM	24
0			BGDESC	49
0			BGCUST	55
0			BGNAME	80
0			BGTAX	82P
0			BGCOD	83
0			BGTAXC	84
0				
0			BGPAID	85
0			BGCPST	86
0			BGIPST	87
0			BGBLNK	92
0			BGINVC	93
0			BGDL0C	95
0			BGGEOC	97
0			BGQTY	100P
0			BGPRIC	103P
0			BGCOST	106P
0			BGPRPG	109P
0			BGMPCT	111P
0			BGFPCT	113P
0			BGEXTN	117P
0			BGRTAX	121P
0			BGSTAX	124P
0			BGTOTL	128P
0	E	82		
0			BGRCOD	2
0			BGREFN	10
0			BGDATE	16
0			BGITEM	24
0			BGDESC	49
0			BGCUST	55
0			BGNAME	80
0			BGTAX	82P
0			BGCOD	83
0			BGTAXC	84
0			BGPAID	85
0			BGCPST	86
0			BGIPST	87
0			BGBLNK	92
0			BGINVC	93

```

0          BGDLOC      95
0          BGGE0C      97
0          BGQTY       100P
0          BGPRIC      103P
0          BGCOST      106P
0          BGPRPG      109P
0          BGMPC      111P
0          BGFPCT      113P
0          BGEXTN      117P
0          BGRTAX      121P
0          BGSTAX      124P
0          BGTOTL      128P
**          MSG1
FUNCTION KEY INVALID AT THIS TIME-TRY AGAIN.
**          MSG2 (ALL BLANKS FOR CLEARING MESSAGE WINDOW)

**          MSG3

MASTER RECORD ADDED - HIT ENTER KEY TO CONTINUE
**          MSG4
RECORD IS NOT IN FILE-HIT F1(ADD) OR RETRY
**          MSG5
NO MORE INVOICE LINES AVAILABLE-REKEY #
**          MSG6
UPDATE RECORD NOT IN FILE-HIT F8 TO EXIT UPDATE SUBR
**          MSG7
MASTER RECORD NOT IN FILE-EXIT SUBROUTINE
**          MSG8
ID# NOT IN GDID FILE.
**          MSG9
NO MORE PMT LINES ON THIS CR MEMO-KEY IN NEW NUMBER
**          MSG10
ID # IS BLANK-KEY DESCRIPTION ON NEXT SCREEN
**          MSG11
DELIVERY LOCATION IS BLANK-KEY IN LOCATION
**          MSG12
QTY X PRICE NE EXTN-PRESS ENTER TO CONTINUE
**          MSG13
SALES/ROAD TAX SHOULD NOT BE ON SAME TICKET
**          MSG14
TOTAL NOT CORRECT-PRESS ENTER TO CONTINUE

```

*Biographical Sketch of Christopher J. Colburn G.D. Armstrong Company,  
Inc. Laytonsville, Maryland*

*Christopher J. Colburn, is a programmer/analyst presently employed by the G.D. Armstrong Company of Laytonsville, Maryland. He "got his start" in programming while working as an operator/programmer with business applications (in RPG) on an IBM System 34 at the Leisure World of Maryland Corporation from 1978 to 1981. In May of 1980, he received a Bachelor of Science degree in Business Administration from the University of Maryland. From 1981 to 1982, he worked at Computer Applications and Systems, Inc., a local computer consulting firm. There he created new and modified existing programs in Accounts Payable, Accounts Receivable, and Job-costing business applications on the HP3000, using both RPG and COBOL languages.*

*In January of 1983, he accepted his present position of programmer/analyst with G.D. Armstrong Company, Inc., a fuel oil and automotive accessory distributor. His accomplishments at Armstrong include: conversion of existing Payroll and A/R programs from an IBM System 32 to the HP3000, creation of an automotive parts inventory system, and development of an A/P-Inventory-A/R database system for use in Armstrong's day-to-day business operations.*