N O T I C E

THIS DOCUMENT HAS BEEN REPRODUCED FROM MICROFICHE. ALTHOUGH IT IS RECOGNIZED THAT CERTAIN PORTIONS ARE ILLEGIBLE, IT IS BEING RELEASED IN THE INTEREST OF MAKING AVAILABLE AS MUCH INFORMATION AS POSSIBLE

80-10179

T77-13973 NASA

JSC-12655

NASA CR-
160646

FINAL DESIGN SPECIFICATION
FOR
ERIPS FIELDS DATA BASE DECK CONVERSION

Job Order 81-127

(E80-10179) FINAL DESIGN SPECIFICATION FOR ERIPS FIELDS DATA BASE DECK CONVERSION (Lockheed Electronics Co.) 41 p HC A03/MF A01 CSCL 05B G3/43

N80-28780

Unclas 00179

Prepared By

Lockheed Electronics Company, Inc.
Aerospace Systems Division
Houston, Texas

Contract NAS 9-15200

FOR

EARTH OBSERVATIONS DIVISION

National Aeronautics and Space Administration

LYNDON B. JOHNSON SPACE CENTER

Houston, Texas

August 1977

LEC-10960

JSC-12655

FINAL DESIGN SPECIFICATION
FOR
ERIPS FIELDS DATA BASE DECK CONVERSION

Job Order 81-127

PREPARED BY

Cheevon Bo-Linn

APPROVED BY

John M. Allred, Supervisor
Physical Sciences Section

P. L. Krumm, Supervisor
Applications Software Section

W. J. Reicks, Manager
Applied Mechanics Department

Prepared By

Lockheed Electronics Company, Inc.

For

Earth Observation Division

Space and Life Sciences Directorate

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS

August 1977

LEC-10960

CONTENTS

Section		Page
1.	SCOPE	1-1
2.	APPLICABLE DOCUMENTS	2-1
3.	SYSTEM DESCRIPTION	3-1
	3.1 HARDWARE DESCRIPTION	3-1
	3.2 SOFTWARE DESCRIPTION	3-1
	3.2.1 SOFTWARE COMPONENT NO. 1 (FDBCVT)	3-1
	3.2.1.1 Linkages	3-1
	3.2.1.2 Interfaces	3-1
	3.2.1.3 Inputs	3-2
	3.2.1.4 Outputs	3-3
	3.2.1.5 Storage Requirements	3-5
	3.2.1.6 Description	3-5
	3.2.1.7 Flowcharts	3-6
	3.2.1.8 Program Listing	3-6
	3.2.2 SOFTWARE COMPONENT NO. 2 (FIND)	3-7
	3.2.2.1 Linkages	3-7
	3.2.2.2 Interfaces	3-7
	3.2.2.3 Inputs	3-8
	3.2.2.4 Outputs	3-8
	3.2.2.5 Storage Requirements	3-8
	3.2.2.6 Description	3-8
	3.2.2.7 Flowcharts	3-8
	3.2.2.8 Program Listing	3-8
	3.2.3 SOFTWARE COMPONENT NO. 3 (NXTCHR)	3-9

Section Page

3.2.3.1 Linkages 3-9
3.2.3.2 Interfaces 3-9
3.2.3.3 Inputs 3-10
3.2.3.4 Outputs 3-10
3.2.3.5 Storage Requirements 3-10
3.2.3.6 Description 3-10
3.2.3.7 Flowcharts 3-10
3.2.3.8 Program Listing 3-10
3.2.4 SOFTWARE COMPONENT NO. 4 (FIXNUM) 3-10
3.2.4.1 Linkages 3-11
3.2.4.2 Interfaces 3-11
3.2.4.3 Inputs 3-11
3.2.4.4 Outputs 3-11
3.2.4.5 Storage Requirements 3-12
3.2.4.6 Description 3-12
3.2.4.7 Flowcharts 3-12
3.2.4.8 Program Listing 3-12
4. OPERATION 4-1
4.1 USER DOCUMENTATION 4-1
4.1.1 PROGRAM SET-UP AND EXECUTION 4-2
4.1.1.1 Terminal Set-Up 4-3
4.1.1.2 Data Deck Input 4-4
4.1.1.3 Printer and Punched Card Output 4-5
4.1.1.4 Terminal Sign-Off 4-7

Section Page

5. TEST PROCEDURE 5-1

5.1 DESCRIPTION OF TEST 5-1

Appendix

A. PROGRAM LISTING. A-1

B. PROGRAM VERIFICATION INPUT AND OUTPUT. B-1

1. SCOPE

1.1 GENERAL

This specification establishes the design of a computer program which converts an ERIPS (Earth Resources Interactive Processing System) Fields Data Base (FDB) update card deck to a card deck compatible with input requirements of the Univac 1108 EOD-LARSYS system.

The Requirement Specifications for the program were provided by the Research, Test, and Evaluation (RT&E) Branch of the Earth Observations Division (EOD) of the National Aeronautics and Space Administration, Lyndon B. Johnson Space Center (NASA/JSC).

2. APPLICABLE DOCUMENTS

The following documents, of exact issue shown, form a part of the specification to the extent herein specified.

- Requirements Specification: REF: Interdepartmental Communication 643-2042.
- IDSD CATEGORY 1 Job Order 81-127, Task Agreement 77-1.
- Section 11, Large Area Crop Inventory Experiment (LACIE) ERIPS User's Guide, Volume 1.

3. SYSTEM DESCRIPTION

3.1 HARDWARE DESCRIPTION

Not applicable

3.2 SOFTWARE DESCRIPTION

The purpose of the program is to input the ERIPS (Earth Resources Interactive Processing System) Fields Data Base (FDB) update card deck and to output (punch) a field definition card deck in the format compatible with the input requirements of the Univac 1108 EOD-LARSYS system of image data processors.

The program is coded in the IBM 360 Fortran IV language, and is executable from the LARS/Purdue (Laboratory for Application of Remote Sensing) terminal in JSC Building 17.

3.2.1 SOFTWARE COMPONENT NO. 1 (FDBCVT)

FDBCVT is the main program. The function of FDBCVT is to read the ERIPS Fields Data Base update card deck and to punch an EOD-LARSYS compatible field definition deck for each field defined in the ERIPS card deck. FDBCVT allows for an optional user-input line and/or sample bias to be applied to the input vertex coordinates of each field of a given sample segment in the ERIPS deck, before punching the output EOD-LARSYS field definition deck(s) for the given sample segment.

3.2.1.1 Linkages

FDBCVT calls three subprograms – FIND, NXTCHR, and FIXNUM – to decode the keywords and parameters of the input ERIPS deck.

3.2.1.2 Interfaces

The program is accessed via the LARS/Purdue terminal in JSC Building 17. The interface between the program and the user is

the LARS/Purdue IBM 360-67 Control Program (CP) and an associated operating system, the Cambridge Monitor System (CMS). The program-user will utilize the terminal keyboard in Building 17 to communicate the appropriate commands to initiate program execution. Operational instructions are provided in section 4.0 of the Final Design Specifications.

The card reader/punch adjacent to the terminal in Building 17 is the program's primary input/output interface.

3.2.1.3 Inputs

The inputs to the Fields Data Base Deck Conversion program, FDBCVT, consist of an optional BIAS card for each sample segment and an ERIPS Fields Data Base update card deck. The format of the ERIPS deck is given in Section 11, ERIPS User's Guide, Volume 1.

The format of the optional BIAS card is:

CC1	CC11	
BIAS	S=XX	L=YY

The parameters "S=XX" and L=YY" on the BIAS card contain the user-supplied integers, "XX" and/or "YY," which are additive sample (S) and/or line (L) bias values to be applied to the input ERIPS deck field coordinates.

The BIAS card is optional. If not input, the defaults used by the program are S=0, L=0. Either S or L or both may be input on the BIAS card.

The input ERIPS FDB update deck is the card deck which normally is output (punched) at the LARS/Purdue terminal in Building 17 using the Del-Foster "DEAF" deck as input to a LARS/Purdue program which provides the ERIPS FDB deck as output.

The key words in the ERIPS FDB deck which are expected and responded to by the conversion program, FDBCVT, are:

SEGSTART - marks the beginning of a set of inputs to be associated with the current sample segment.

FLDSTART - marks the beginning of a field definition card

FIELD - contains the parameters that define the current field

CLASS - identifies the category/class/subclass for the current field

LINEXX (where XX are numeric) - defines the line coordinate of the field's vertex

PIXELXX (where XY are numeric) - defines the pixel coordinate of the field's vertex

FLDEND - marks the end of a set of field definition cards

SEGEND - marks the end of the input cards for the current sample segment

Any other key words present in the ERIPS deck are ignored by FDBCVT.

3.2.1.4 Outputs

The FDB deck conversion program, FDBCVT, provides both line printer and card punch output.

Primary output is the punched cards in a format compatible with the Univac 1108 EOD-LARSYS input requirements. The punched card output consist of cards in the following formats:

Card type	CC1	CC11
Comment card	COMMENT	SAMPLE SEGMENT ICCCC
Class name card	CLASSNAME	CNAME
Field definition card	FNAME	(1,1), (XXX,YYY), (XXX,YYY), (XXX,YYY), (XXX,YYY), *
Field definition continuation card		,(XXX,YYY), (XXX,YYY),···

FNAME is the field name (1-6 alphanumeric characters - first character must be alphabetic) read from the input FLDSTART card. Printer output provided by the program is as follows:

1. An optional print-out of the input deck.
2. An optional print-out of the output (punched) deck with possible error messages.
3. The error messages are as follows:
   a. If an input SEGSTART card cannot be paired with a SEGEND card, the message is:

      "ERROR--A VALID SEGSTART (SEGEND) CARD BEFORE SEGSTART ID=ICCCC IS MISSING."

   b. If the input SEGSTART card is incorrectly formatted (does not have the "=" following "ID") the message is:

      "ERROR--THE SEGSTART CARD (CURRENT SEGSTART CARD) IS MISSING AN EQUALS SIGN--LOOK FOR THE NEXT SEGSTART OR EOF."

   c. If an input FLDSTART card cannot be paired with a FLDEND card, the message is:

      "ERROR--A VALID FLDSTART (FLDEND) CARD BEFORE FLDSTART NAME=CCCCCC IS MISSING."

   d. If an input FLDSTART card is incorrectly formatted (does not have the "=" following "NAME" the message is:

      "ERROR--THE FLDSTART CARD (CURRENT FLDSTART CARD) IS MISSING AN EQUALS SIGN--LOOK FOR THE NEXT FLDSTART OR SEGEND CARD."

   e. If, on the input FIELD cards, each pixel coordinate cannot be paired with its correct line coordinate or vice versa, the message is:

      "ERROR--FOR FIELD CCCCCC THE NUMBER OF PIXELS DOES NOT MATCH WITH THE NUMBER OF LINES."

   f. If, on the input BIAS card, an "=" is not found following either "S" or "L," the message is:

"ERROR IN BIAS CARD--THE EQUALS SIGN IS MISSING FOR EITHER THE SAMPLE AND/OR LINE INCREMENT."

g. When reading the line/pixel coordinates from the FIELD cards, if a non-numeric is encountered in a position where a numeric digit is expected (i.e., in the positions occupied by XX or YY in LINEXX = YY or PIXELXX = YY) the message is:

"***CARD IN ERROR IS - FIELD LINEXX = YY
PIXELXX = YY..."

3.2.1.5 Storage Requirement

The program requires 8080 bytes of storage.

3.2.1.6 Description

The program reads the ERIPS Fields Data Base update deck, card-by-card. The deck may include a user-supplied BIAS card preceding a SEGSTART card. The sample (S) and/or line (L) bias value following the "=" will be added to each input sample and/or line coordinate given on the FIELD card(s) for the given sample segment. The sample/line bias is initialized to zero (0) at the beginning of the program, and at each SEGEND card encountered in the input ERIPS deck. This requires the BIAS card to be present, preceding a SEGSTART card, in order for bias values to be applied to the input field coordinates for a given sample segment. The values input on a BIAS card are added to each of the sample and line coordinates for all fields defined between a SEGSTART card and the associated SEGEND card.

For each "SEGSTART ID=ICCCC" card read, the program punches a LARSYS comment card, "COMMENT SAMPLE SEGMENT ICCCC."

For each "FLDSTART NAME=FNAMEX" card read, the field name following "NAME=" will be the name placed in columns 1-6 of the output definition cards.

For each set of "FIELD CLASS=CNAMEA LINE01=XX PIXEL01=YY LINE02=XX PIXEL02=YY..." cards read following the "FLDSTART" card and preceding a FLDEND" card, the program outputs a LARSYS "CLASSNAME CNAMEA" card, followed by EOD-LARSYS field definition cards with the field name (columns 1-6) from the input FLDSTART card. The output field coordinates include the bias value(s) from the BIAS card, if input. The format of the output field definition cards is given in Section 3.2.1.4.

The program continues to read cards from an input ERIPS deck until an end-of-file is encountered.

The punched cards output by the program are in the Univac FIELDATA character set (i.e., any necessary conversion of punched card codes for characters from IBM EBCDIC to Univac FIELDATA is provided by the program).

The format of the input ERIPS deck is expected to be in the format described in the ERIPS User's Guide, Volume 1, Section 11. The program provides error messages if problems are encountered in interpreting the keywords, separators, or parameters on the input cards. The error conditions and resulting printed messages are described in Section 3.2.1.4.

3.2.1.7 Flowcharts

Not applicable.

3.2.1.8 Program Listing

See Appendix A.

3.2.2 SOFTWARE COMPONENT NO. 2 (FIND)

The purpose of the subprogram, Function FIND, is to perform a search for a specific character.

3.2.2.1 Linkages

Function FIND is called by the main program, FDBCVT. Function FIND does not reference any other subprograms.

3.2.2.2 Interfaces

Function FIND interfaces with the calling program via three calling arguments and the function value, which is set within Function FIND.

The function value is set = 1, if a successful character search is completed.

The function value is set = -1 if the character search is unsuccessful.

The calling arguments for FUNCTION FIND are:

ARGUMENT	DIMENSION	TYPE	IN/OUT	DESCRIPTION
CARD	68	A	IN	The input array of 68 words which is assumed to have one character per word, left-justified, blank-filled.
COL	1	I	IN/OUT	On input, the location (word) in CARD, preceding the location at which the search is to begin. On output, the location in CARD at which the character was found. If the character is not found in CARD, COL = initial input value.
VECTOR	1	A	IN	Contains the character to be searched for, left-justified blank-filled in the word.

3.2.2.3 Inputs

The inputs to Function FIND are three calling arguments – CARD, COL, VECTOR – described in Section 3.2.2.2.

3.2.2.4 Outputs

Output from Function FIND is via one calling argument, COL, and the function value which is set within the subprogram (see section 3.2.2.2).

3.2.2.5 Storage Requirements

Function FIND requires 514 bytes of storage.

3.2.2.6 Description

Function Find performs a search of an input (argument) array, CARD, for the alphanumeric character given in the input argumer VECTOR. The search in CARD will begin at the next location in CARD following the location specified in the input argument, COL. When the specified character is located in CARD, the function value is set equal to 1, and the location of the character position in CARD is returned in COL. If the search for the specified character is unsuccessful, the function value is set equal to -1, and COL is returned containing the value it had on entry to Function Find.

3.2.2.7 Flowcharts

Not applicable.

3.2.2.8 Program Listing

See Appendix A.

3.2.3 SOFTWARE COMPONENT NO. 3 (NXTCHR)

The purpose of the subprogram, FUNCTION NXTCHR, is to scan a given vector for a non-blank alphanumeric character.

3.2.3.1 Linkages

The subprogram, Function NXTCHR, is referenced by the main program, FDBCVT. The subprogram does not reference any other subprograms.

3.2.3.2 Interfaces

Function NXTCHR interfaces with the calling program via two calling arguments and the function value, which is set within the subprogram.

The function value returned is an alphanumeric character. The character returned is either the first non-blank character found in the input array, CARD, or a "blank" if a non-blank character is not located in CARD.

The calling arguments for Function NXTCHR are:

Argument	Dimension	Type	In/out	Description
CARD	68	A	In	An input array of characters, one character per word, left-justified and blank-filled in each word.
COL	1	I	In/out	On input, COL = the location in CARD preceding the location at which the search for the next non-blank character is to begin. On output, either COL = the location in CARD at which a non-blank character was found, or COL = 67 (the maximum size -1 of CARD) if CARD was all blanks.

3.2.3.3 Inputs

The inputs to Function NXTCHR are two calling arguments – CARD and COL – described in Section 3.2.3.2.

3.2.3.4 Outputs

The output from Function NXTCHR is via the function value and one calling argument, COL (see Section 3.2.3.2).

3.2.3.5 Storage Requirements

Function NXTCHR requires 478 bytes of storage.

3.2.3.6 Description

Function NXTCHR performs a search of an input (argument) array, CARD, for a non-blank alphanumeric character. The search in CARD will begin at the next location in CARD following the location specified in the input argument, COL. When a non-blank alphanumeric character is found in CARD, the function value is set equal to the character found, and the location (in CARD) of the character is returned in COL. If a non-blank character is not located in CARD, the function value returned is "blank," and COL = 67 (the maximum size -1 of CARD).

3.2.3.7 Flowcharts

Not applicable.

3.2.3.8 Program Listing

See Appendix A.

3.2.4 SOFTWARE COMPONENT NO. 4 (FIXNUM)

The purpose of the subprogram, Function FIXNUM, is to convert an EBCDIC numeric character to an integer digit.

3.2.4.1 Linkages

Function FIXNUM is called by the main program, FDBCVT. Function FIXNUM does not reference any other subprograms.

3.2.4.2 Interfaces

Function FIXNUM interfaces with the calling program via two calling arguments and the function value, which is set within Function FIXNUM.

The function value returned is the integer resulting from the conversion of the EDCDIC character.

The calling arguments for Function FIXNUM are:

Argument	Dimension	Type	In/out	Description
NUM	1	A	IN	NUM contains the EBCDIC character, left-justified in the word.
MASK	1	A	IN	MASK contains the EBCDIC numeric character "0" (zero), right justified and sign-filled.

3.2.4.3 Inputs

The inputs to Function FIXNUM are two calling arguments — NUM and MASK — described in Section 3.2.4.2, above.

3.2.4.4 Outputs

The only output of Function FIXNUM is via the function value that is set within FIXNUM. The function value returned to the calling program is the integer which results from the conversion of an EBCDIC numeric character.

3.2.4.5 Storage Requirements

Function FIXNUM requires 584 bytes of storage.

3.2.4.6 Description

Function FIXNUM converts one EBCDIC numeric character input in the calling argument NUM. The conversion of the EBCDIC character to an integer digit is as follows:

1. The input character in NUM is shifted to the right 24 binary positions, resulting in the character being right-justified and the remainder of the word sign-filled (all binary 1's).

2. The right-justified, sign-filled value in MASK (an EBCDIC zero) is subtracted from the right-justified, sign-filled value in NUM.

   The result of the subtraction is an integer, in the range 0-9, if the EBCDIC character in NUM is one of the set, "0", "1", "2", ..., "9".

3. The result of the subtraction is returned as the function value.

   If the result of the subtraction is not an integer in the range 0-9, FIXNUM also outputs a printed message

   "*** ERROR - NUMERIC CHARACTER EXPECTED AND NOT FOUND!"

3.2.4.7 Flowcharts

Not applicable.

3.2.4.8 Program Listing

See Appendix A.

4. OPERATION

FDBCVT is executed on the LARS/PURDUE IBM 360/67 computer using the remote terminal facilities in JSC Building 17. Program operation is described in terms of the terminal operations necessary to execute the program from Building 17.

4.1 USER DOCUMENTATION

The main Fortran IV Program, FDBCVT, along with the subroutines FIND, NXTCHR, and FIXNUM have been placed in a permanent disk file which is referenced by an ID and password provided by the Research, Test, and Evaluation Branch (RT&E). The program can be called from the Hazeltine 2000 terminal or from the 2741 Typewriter terminal by typing in the name of the main program, "FDBCVT".

The following capabilities are provided to the user after the data deck is read in via the card reader and referenced to the program, FDBCVT:

1. Obtain a listing of the 'FDBCVT' input data deck,
2. Obtain a listing of the punched output card deck along with any error messages pertaining to the input data deck,
3. The punched cards output by FDBCVT, via the card punch adjacent to the terminal, formatted for input to the Univac 1108 EOD-LARSYS program.

Program set-up and use instructions are provided below.

4.1.1 PROGRAM SET-UP AND EXECUTION

The input cards [an ERIPS Fields Data Base Update deck] must include an additional card, supplied by the user. The required first card of the input deck is of the format:

CC1	CC10	
ID	JSC200	(RT&E account ID at LARS/PURDUE)

The "ID" card is a LARS/Purdue system requirements, to associate the input with the correct terminal user.

The order of activities for program executions are:

1. Terminal sign-on (LOG IN), and acquire temporary file space for program execution.
2. Transmit input card deck to Purdue.
3. Execute FDBCVT.
4. Initiate print-out (if needed).
5. Initiate card punch output.
6. Retrieve print-out and punched cards.
7. Log out, on the terminal.
8. Interpret the punched cards (on 026 keypunch machine).

The sequence of terminal activities below are for the Hazeltine 2000 terminal.

NOTE (1) In the sequence of terminal commands and responses given below, the caret (">") indicates the required user-type in, the brackets "[ ]" indicate system response. The ">" is displayed by the system, to elicit user-input. The brackets are for documentation convenience only.

NOTE (2) On the Hazeltine 2000 terminal, the user-command is transmitted by depressing the carriage return ("CR") key.

On the 2741 terminal, the user-command is transmitted by "RETURN" key.

NOTE (3) On the Hazeltine 2000 terminal, to erase a typed-in character, the "@" key is depressed.

To erase an entire typed-in line, the "[" key is depressed.

NOTE (4) On the 2741 terminal, to erase a typed-in character the "@" key is depressed.

To erase an entire typed-in line, the "¢" key is depressed.

4.1.1.1 Terminal Set-Up

On the Hazeltine 2000 terminal, make sure that the green box closest to your terminal is switched to 'LARS'.

User: Depress 'CR' (on the 2741 terminal, depress "ATTN"). If the terminal does not respond back with 'RESTART', type in 'L∅JSC200', depress 'CR'.

Terminal: [RESTART]

User: >L JSC200
depress 'CR'

Terminal: [ENTER PASSWORD]

User: > "ABC"
depress 'CR'

(NOTE: The actual password to be used in place of "ABC" is the password allocated to RT&E associated with the account ID, "JSC200".)

Terminal: [ENTER NAME]
> (TYPE IN YOUR INITIALS OR NAME)
depress 'CR'
[YOUR OPERATORS ARE ...]

```
[CP]
> I CMS
[CMS READY]
> DISX SET S (request for small (="S") temporary file)
[LINE AND CHARACTER SET TO 1]
[YOU ARE LINKED TO TEMP DISK XX]
[P(192):  XX FILES; YYY REC IN USE, ZZ LEFT (OF 296),
 XY% FULL (X CYL)]
```

The status of the disk's storage space is obtained as follows:

```
> LISTF
depress 'CR'
[FILENAME FILETYPE MODE ...]
```

If more storage space is needed than is currently available on the temporary disk file, the temporary file may be "cleaned up". To erase files in order to increase the amount of storage space, type in:

```
> ERASE (type in one of the listed filename) (type in the file-
name's filetype)
depress 'CR'
```

Continue the above process of erasing files from the temporary disk until enough storage space is available on the disk to handle the execution of FDBCVT. All printer output of the program is stored on this file.

4.1.1.2 Data Deck Input

1. Proceed to the card reader, adjacent to the terminal.
2. If any reading, printing, or punching is in progress, wait until the operation is completed.
3. On the card reader/punch control panel, depress the 'NPRO' button.

4. Put the "ID JSC200" card on top of the data deck.

5. Place the input cards in the card-hopper FACE DOWN, "9-edge leading" – i.e.,) with the top edge of the cards facing outward.

6. Place the card weight on top of the DECK.

7. a. On the card reader/punch control panel, turn the knob to 'TSM TRSP'.

   b. Depress the 'EOF' button

   c. Depress the 'START' button, hold until the 'READY' light goes on.

8. After all of the cards have been read in, an audible beeping sound will be generated, signifying that the transmission is complete.

9. Depress the 'NPRO' button.

10. Turn the knob to 'OFF-LINE'

11. Remove the input card deck from the card reader hopper.

12. Return to the terminal console – the input deck is now available to program FDBCVT.

4.1.1.3 Printer and Punched Card Output

After a few seconds, depress 'CR'.

```
[** CARDS XFERED BY HOUSTON ...]

>0 READ FDBCVT DATA
 depress 'CR'

[R, T = ...]

>FDBCVT
 depress 'CR'

[XX.YY.ZZ   FILEDEF 5 DSK-P1 ...]
```

If an off-line copy of printer output is needed (with possible error messages), type in:

>0 PRINT PRINT LISTING

NOTE (A): Do NOT depress 'CR' if any card reading, card punching, or printing is taking place at this time, by other terminal facility users. Wait until the terminal input/output activities (card reader and printer) are not being used, then depress 'CR'to send the "PRINT" request. When an audible beeping sound is generated, LARS is attempting to transmit the requested printout.

1. Proceed to the printer and turn the knob to 'PRINT'
2. Depress the 'START' key on the printer control panel
3. When the printing has stopped, depress 'CARRIAGE STOP' then 'CARRIAGE RESTORE' (= paper feed)

If a printer listing of the input data cards is needed, type in:

>0 PRINT FDBCVT DATA

See NOTE (A), before depressing 'CR'

To get the output cards punched, type in:

>0 PUNCH PUNCH OUTPUT

See NOTE (A), before depressing 'CR'

Proceed to the card reader:

1. Wait for a beeping sound to be generated.
2. Turn the knob to 'PUNCH', on the card reader control panel.
3. Place blank cards in the card reader, "9-edge leading".
4. a. Depress the 'START' button, hold until the 'READY' light goes on.

   b. Card punching should begin when the 'READY' light goes on.

5. When the beeping sound is generated, remove the unused blank cards from the card reader hopper.

6. Depress the 'NPRO' button.

7. Turn the knob to 'OFF-LINE'.

8. Remove the punched cards from the card hopper and strip out any leading or trailing blank cards.

9. Interpret the punched deck on the '026' keypunch machine.

4.1.1.4 Terminal Sign-Off

User: Depress the 'BREAK' key (to get from CMS to CP)

Terminal: [CP]

User: >Logout

[CONNECT = XX:YY:ZZ VIRTCPU = XXX:YY.ZZ

TOPCPU = XXX:YY.ZZ]

[LOGOUT AT XX.YY.ZZ ON MM/DD/YY]

[CP-67 ONLINE]

5. TEST PROCEDURE

5.1 DESCRIPTION OF TEST

Using representative input cards from an ERIPS FDB deck, the program was executed from the terminal in JSC Building 17. The input deck also included simulated ERIPS FDB cards with erroneous parameters, in order to test the diagnostic error messages incorporated in the program. The run was executed to verify

a. The punched card output, in EOD-LARSYS input format.

b. The optional print-out of input cards and any error diagnostics.

The input and output of the verification run of the program is in Appendix B of this document.

TEST VERIFICATION

For ERIPS FIELDS DATA BASE DECK CONVERSION

This verification is being conducted to insure that the delivered program products satisfy the requirements as originally stated by the requesting organization.

R. P. Heydorn
NASA Monitor

Thomas C. Minter
Requestor

Cheeven Bob
Developer

P J Aaron Jr
Cognizant System Manager

[redacted] 7-5-77
Quality Assurance

Cheeven Bob
Test Conductor

Verification Date: 7-5-77

APPENDIX A

PROGRAM LISTING

FILE FDBCVT

```
0001                IMPLICIT INTEGER (A-H,O-Z)                                      FDR00010
0002                DIMENSION KEY(6),CHNG(3),FLDNM(6),CLNM(6),LN(10),PX(10),MI(4),  FDR00020
                   1 CONT(2),CARD2(68)                                              FDR00030
0003                DIMENSION SCRPX(10),SCRLN(10)                                   FDR00040
0004                REAL MINLN,MAXPX,CELN,CEPX,GTLN,GTPX,CF,GT,CFSIN,GTSIN,         FDR00050
                   #      CFAB,GTAB ,CFY,GTY                                       FDR00060
0005                REAL CFRP,CFRL,GTRP,GTRL                                        FDR00070
0006                DATA KEY/'SEGS','SEGE','FLDS','FLDE','FIEL','BIAS'/            FDR00080
0007                DATA SEP/'='/,BLK/' '/,CHNG/'L','N','E'/,CONT/'*',' '/          FDR00090
0008                DATA CLP/Z6B6C4040/,LP/Z6C404040/,RP/Z4C404040/                 FDR00100
0009                 DATA SSS/'S'/                                                  FDR00110
0010                DATA MASK/'0'/                                                  FDR00120
0011                MASK = MASK * (2.0**(-24.0))                                    FDR00130
0012                BIKEY=0                                                         FDR00140
0013                SERR = 0                                                        FDR00150
0014                FERKEY = 0                                                      FDR00160
0015                FSTKEY = 0                                                      FDR00170
0016                FERR = 0                                                        FDR00180
0017                SSTKEY = 0                                                      FDR00190
0018                SEDKEY = 0                                                      FDR00200
0019                PXFLAG = 0                                                      FDR00210
0020                SKFLAG = 0                                                      FDR00220
0021                LNFLAG = 0                                                      FDR00230
0022          19    READ (5,10,END=5002) CODE,CARD2                                 FDR00240
0023          10    FORMAT (A4,68A1)                                                FDR00250
0024                COL = 0                                                         FDR00260
                  C*                                                              FDR00270
                  C*    IF A FLDSTART CARD IS IN ERROR, LOOK FOR THE NEXT FLDSTART OR FDR00280
                  C*    SEGEND CARD                                               FDR00290
                  C*                                                              FDR00300
0025                IF (FERR.NE.1) GO TO 17                                         FDR00310
0026                DO 16 K=2,3                                                     FDR00320
0027                IF(KEY(K).NE.CODE)GO TO 15                                      FDR00330
0028                GO TO (19,40,50),K                                              FDR00340
0029             15 CONTINUE                                                        FDR00350
0030             16 CONTINUE                                                        FDR00360
0031                GO TO 19                                                        FDR00370
                  C*                                                              FDR00380
                  C*    IF A SEGSTART CARD IS IN ERROR, LOOK FOR THE NEXT SEGSTART OR EOF FDR00390
                  C*                                                              FDR00400
0032          17    IF (SERR.NE.1) GO TO 18                                         FDR00410
0033                IF (KEY(1).NE.CODE) GO TO 19                                    FDR00420
0034                SERR = 0                                                        FDR00430
0035                GO TO 30                                                        FDR00440
0036          18    DO 20 K=1,6                                                     FDR00450
0037                IF(KEY(K).NE.CODE)GO TO 21                                      FDR00460
0038                GO TO (30,40,50,60,70,80),K                                     FDR00470
0039             21 CONTINUE                                                        FDR00480
0040             20 CONTINUE                                                        FDR00490
0041                GO TO 19                                                        FDR00500
                  C*                                                              FDR00510
                  C*    SEGSTART                                                  FDR00520
                  C*                                                              FDR00530
0042          30    M=FIND(CARD2,COL,SEP)                                           FDR00540
0043                IF (M.NE.1) GO TO 37                                            FDR00550
0044                COL = COL + 1                                                   FDR00560
0045                DO 31 I=1,4                                                     FDR00570
0046                COL = COL+1                                                     FDR00580
0047                MI(I)=FIXNUM(CARD2(COL),MASK)                                   FDR00590
0048                IF(MI(I).LT.0.OR.MI(I).GT.9) GO TO 4000                         FDR00600
0049             31 CONTINUE                                                        FDR00610
0050                M = MI(1)*1000 + MI(2)*100 + MI(3)*10 + MI(4)                   FDR00620
0051                WRITE (6,32) M                                                  FDR00630
0052             32 FORMAT(5X,'COMMENT',6X,'SAMPLE SEGMENT  I',I4)                  FDR00640
0053                PUNCH 38,M                                                      FDR00650
0054             38 FORMAT('COMMENT',6X,'SAMPLE SEGMENT  I',I4)                     FDR00660
                  C*                                                              FDR00670
                  C*    CHECK TO SEE IF EACH SEGSTART CARD CAN BE PAIRED WITH A SEGEND FDR00680
                  C*                                                              FDR00690
0055                SSTKEY = SSTKEY + 1                                             FDR00700
0056                J = SEDKEY + 1                                                  FDR00710
0057                IF (SSTKEY.EQ.J) GO TO 19                                       FDR00720
0058                IF (SSTKEY.LT.J) GO TO 34                                       FDR00730
                    WRITE (6,33) CODE,CARD2                                       FDR00740
0059             33 FORMAT(//5X,'ERROR---A SEGEND CARD BEFORE',2X,A4,2X,68A1/5X,'IS MISSING'FDR00750
                   1//)                                                           FDR00760
```

FILE FDBCVT

```
0061                    SEGEND = SSTKEY - 1                                                 FDB00770
0062                    FERR=0                                                              FDB00780
0063                    GO TO 19                                                            FDB00790
0064          34        WRITE (6,35) CODE,CARD2                                             FDB00800
0065          35        FORMAT(//5X,'ERROR--A VALID SEGSTART CARD BEFORE'/3X,A4,68A1/5X,'   FDB00810
                       1IS MISSING')
0066                    SSTKEY = SEDKEY + 1                                                 FDB00820
0067                    GO TO 19                                                            FDB00830
0068          36        WRITE (6,37) CODE,CARD2                                             FDB00840
0069          37        FORMAT(//5X,'ERROR--THE SEGSTART CARD'/3X,A4,68A1/5X,'IS MISSING    FDB00850
                       1AN EQUALS SIGN--LOOK FOR THE NEXT SEGSTART OR EOF')                FDB00860
0070                    SERR = 1                                                            FDB00870
0071                    SEDKEY = SEDKEY + 1                                                 FDB00880
0072                    GO TO 19                                                            FDB00890
                C*                                                                          FDB00900
                C*      SEGEND                                                              FDB00910
                C*                                                                          FDB00920
0073          40        SEDKEY = SEDKEY + 1                                                 FDB00930
0074                    FERR = 0                                                            FDB00940
0075                    LINKEY = 0                                                          FDB00950
0076                    SAMPLE = 0                                                          FDB00960
0077                    LINE = 0                                                            FDB00970
0078                    GO TO 19                                                            FDB00980
                C*                                                                          FDB00990
                C*      FLDSTART                                                            FDB01000
                C*                                                                          FDB01010
0079          50        FERR = 0                                                            FDB01020
0080                    LNCNT = 0                                                           FDB01030
0081                    PXCNT = 0                                                           FDB01040
0082                    M = FIND(CARD2,COL,SEP)                                             FDB01050
0083                    IF (M.NE.1) GO TO 51                                                FDB01060
0084                    DO 52 I=1,6                                                         FDB01070
0085                    FLDNM(I)=BLK                                                        FDB01080
0086          52        FLDNM(I) = NXTCHR(CARD2,COL)                                        FDB01090
                C*                                                                          FDB01100
                C*      CHECK TO SEE IF EACH FLDSTART CARD CAN BE PAIRED UP WITH A          FDB01110
                C*      FLDEND CARD                                                         FDB01120
                C*                                                                          FDB01130
0087                    FSTKEY = FSTKEY + 1                                                 FDB01140
0088                    J = SEDKEY + 1                                                      FDB01150
0089                    IF (FSTKEY.EQ.J) GO TO 19                                           FDB01160
0090                    IF (FSTKEY.GT.J) GO TO 53                                           FDB01170
0091                    WRITE (6,54) CODE,CARD2                                             FDB01180
0092          54        FORMAT(//5X,'ERROR--A VALID FLDSTART CARD BEFORE'/3X,A4,68A1/5X,    FDB01190
                       1'IS MISSING')                                                      FDB01200
0093                    FSTKEY = FEDKEY + 1                                                 FDB01210
0094                    GO TO 19                                                            FDB01220
0095          53        WRITE (6,55) CODE,CARD2                                             FDB01230
0096          55        FORMAT(//5X,'ERROR--A FLDEND CARD BEFORE'/3X,A4,68A1/5X,'IS MISSIN  FDB01240
                       1G')                                                                FDB01250
0097                    FEDKEY = FSTKEY - 1                                                 FDB01260
0098                    GO TO 19                                                            FDB01270
0099          51        WRITE (6,56) CODE,CARD2                                             FDB01280
0100          56        FORMAT(//5X,'ERROR--THE FLDSTART CARD'/3X,A4,68A1/5X,'IS MISSING    FDB01290
                       1AN EQUALS SIGN--LOOK FOR THE NEXT FLDSTART OR SEGEND CARD')        FDB01300
0101                    FERR = 1                                                            FDB01310
0102                    FEDKEY = FEDKEY + 1                                                 FDB01320
0103                    GO TO 19                                                            FDB01330
                C*                                                                          FDB01340
                C*      FLDEND                                                              FDB01350
                C*                                                                          FDB01360
0104          60        FEDKEY = FEDKEY + 1                                                 FDB01370
0105                    IF (PXCNT.NE.LNCNT) GO TO 62                                        FDB01380
                C*                                                                          FDB01390
                C*                                                                          FDB01400
                C*      THIS BLOCK OF CODE WILL DETERMINE WHETHER THE USER INPUT VERTICES    FDB01410
                C*      IN A CLOCKWISE OR COUNTER CLOCKWISE ORDER.  IF COUNTER CLOCKWISE    FDB01420
                C*      IS DETERMINED, THE ORDER IS CHANGED TO CLOCKWISE. THE ORDER IS      FDB01430
                C*      DETERMINED BY THE ANGLES PRODUCED FROM THE LINES OF THE POINTS      FDB01440
                C*      COMING INTO AND OUT OF THE POINT THAT HAS THE MINIMUM LINE NO.      FDB01450
                C*      IF THE ANGLE OF THE LINE COMING INTO THE MIN. LINE POINT IS .GT.    FDB01460
                C*      THE ANGLE PRODUCED BY THE LINE COMING OUT OF THE MIN. LINE POINT,   FDB01470
                C*      THEN THE ORDER OF VERTICES IS IN CLOCKWISE ORDER.                   FDB01480
                C*                                                                          FDB01490
0106                    IF(PXCNT.LE.2)GO TO 671                                             FDB01500
0107                    OFSET = 0                                                           FDB01510
0108                    MLN= LN(1)                                                          FDB01520
```

A-2
36

FILE FDBCVT

```
0109                MPX = PX(1)                                                  FDB01530
0110                INC = 1                                                      FDB01540
0111                DO 666 I=2,PXCNT                                             FDB01550
0112                FL = LN(I)                                                   FDB01560
0113                IF(MLN.LT.FL)GO TO 665                                       FDB01570
0114                IF(MLN.NE.FL) GO TO 664                                      FDB01580
0115                IF(MPX.LT.PX(I))GO TO 665                                    FDB01590
0116              664 MLN = FL                                                   FDB01600
0117                MPX = PX(I)                                                  FDB01610
0118                INC = I                                                      FDB01620
0119              665 CONTINUE                                                   FDB01630
0120              666 CONTINUE                                                   FDB01640
0121                IF(INC.NE.1)GO TO 667                                        FDB01650
0122                CFSET = 1                                                    FDB01660
0123                CFLN = FLOAT(LN(PXCNT))                                      FDB01670
0124                CFPX= FLOAT(PX(PXCNT))                                       FDB01680
0125                GO TO 668                                                    FDB01690
0126              667 IF(INC.NE.PXCNT) GO TO 668                                 FDB01700
0127                GTLN = FLOAT(LN(1))                                          FDB01710
0128                GTPX = FLOAT(PX(1))                                          FDB01720
0129                GO TO 669                                                    FDB01730
0130              668 GTLN = FLOAT(LN(INC+1))                                    FDB01740
0131                GTPX= FLOAT(PX(INC+1))                                       FDB01750
0132                IF(CFSET.EQ.1) GO TO 670                                     FDB01760
0133              669 CFLN = FLOAT(LN(INC-1))                                    FDB01770
0134                CFPX = FLOAT(PX(INC-1))                                      FDB01780
0135              670 CONTINUE                                                   FDB01790
0136                MINLN = FLOAT(MLN)                                           FDB01800
0137                MINPX= FLOAT(MPX)                                            FDB01810
0138                CFRL = CFLN - MINLN                                          FDB01820
0139                IF(CFRL.LE.0)CFRL = CFRL* (-1)                               FDB01830
0140                CFRP = CFPX - MINPX                                          FDB01840
0141                IF(CFRP.LE.0)CFRP = CFRP * (-1)                              FDB01850
0142                CF = SQRT( (CFRL**2.0) + (CFRP**2.0) )                       FDB01860
0143                GTRL = GTLN - MINLN                                          FDB01870
0144                IF(GTRL.LE.0)GTRL = GTRL * (-1)                              FDB01880
0145                GTRP = GTPX - MINPX                                          FDB01890
0146                IF(GTRP.LE.0)GTRP = GTRP * (-1)                              FDB01900
0147                GT = SQRT( (GTRL**2.0) + (GTRP**2.0) )                       FDB01910
0148                CFY = SQRT(CFRL)                                             FDB01920
0149                GTY = SQRT(GTRL)                                             FDB01930
0150                GTSIN = GTY/GT                                               FDB01940
0151                CFSIN = CFY/CF                                               FDB01950
0152                CFAG = ARSIN(CFSIN)                                          FDB01960
0153                GTAG = ARSIN(GTSIN)                                          FDB01970
0154                IF(CFPX.LT.MINPX)CFAG = 180. - CFAG                          FDB01980
0155                IF(GTPX.LT.MINPX)GTAG = 180. - GTAG                          FDB01990
0156                IF(GTPX.EQ.MINPX)GTAG =90.0                                  FDB02000
0157                IF(CFPX.EQ.MINPX)CFAG=90.0                                   FDB02010
0158                ST = 2                                                       FDB02020
0159                IF(CFAG.GT.GTAG)GO TO 69                                     FDB02030
0160                IF(CFAG.NE.GTAG)GO TO 57                                     FDB02040
0161              671 IF(PX(1).LE.PX(2))GO TO 69                                 FDB02050
0162                ST = 1                                                       FDB02060
                  C*                                                           FDB02070
                  C*    ORDER VERTICES IN CLOCKWISE ORDER                      FDB02080
                  C*                                                           FDB02090
0163                57 STT = PXCNT + 2                                         FDB02100
0164                IF(ST.EQ.1)STT=PXCNT + 1                                   FDB02110
0165                DO 58 N=ST,PXCNT                                           FDB02120
0166                I = STT - N                                                FDB02130
0167                SCRPX(N) = PX(I)                                           FDB02140
0168                SCRLN(N) = LN(I)                                           FDB02150
0169                58 CONTINUE                                                FDB02160
0170                DO 59 I = ST,PXCNT                                         FDB02170
0171                PX(I) = SCRPX(I)                                           FDB02180
0172                LN(I) = SCRLN(I)                                           FDB02190
0173                59 CONTINUE                                                FDB02200
                  C*                                                           FDB02210
                  C*                                                           FDB02220
                  C*    PUNCH THE FIELD DEFINITON CARDS WITH FOUR VERTICES PER CARD  FDB02230
                  C*                                                           FDB02240
0174                69 NTIMES = PXCNT/4                                        FDB02250
0175                IF ((NTIMES*4).NE.PXCNT) NTIMES=NTIMES+1                   FDB02260
0176                IF(BIKEY.NE.1) GO TO 63                                    FDB02270
0177                DO 210 I=1,PXCNT                                           FDB02280
0178                PX(I) = PX(I) + SAMPLE
```

A-3 37

FILE FDBCVT

```
0179              LN(I) = LN(I) + LINE                                                  FDB02290
0180            210 CONTINUE                                                          FDB02300
0181             63 C = CONT(1)                                                       FDB02310
0182                IF (NTIMES.EQ.1) C=CONT(2)                                        FDB02320
0183                IF (PXCNT.LE.4) D=PXCNT                                           FDB02330
0184                IF (PXCNT.GT.4) D = 4                                             FDB02340
0185                WRITE(6,64)(FLDNM(I),I=1,6),LP,RP,(CLP,PX(I),LN(I),RP,I=1,D),C    FDB02350
0186                PUNCH 65 , (FLDNM(I),I=1,6),LP,RP,(CLP,PX(I),LN(I),RP,I=1,D),C    FDB02360
0187             64 FORMAT(//5X,6A1,4X,A1,'1,1',A1,4(A2,I3,',',I3,A1),11X,A1)         FDB02370
0188              65 FORMAT(6A1,4X,A1,'1,1',A1,4(A2,I3,',',I3,A1),11X,A1)             FDB02380
0189                IF (NTIMES.EQ.1) GO TO 19                                         FDB02390
0190                DO 66 I=2,NTIMES                                                  FDB02400
0191                C = CONT(1)                                                       FDB02410
0192                IF (NTIMES.EQ.I) C=CONT(2)                                        FDB02420
0193                D = D + 1                                                         FDB02430
0194                IF (PXCNT.LE.(D+4)) E=PXCNT                                       FDB02440
0195                IF (PXCNT.GT.(D+4)) E=D+4                                         FDB02450
0196                PUNCH 67,  (CLP,PX(I),LN(I),RP,I=D,E),C                           FDB02460
0197                WRITE(6,67)(CLP,PX(I),LN(I),RP,I=D,E),C                           FDB02470
0198              67 FORMAT(10X,5(A2,I3,',',I3,A1),11X,A1)                            FDB02480
0199             66 CONTINUE                                                          FDB02490
0200              73 DO 74 I=1,6                                                      FDB02500
0201                FLDNM(I) = BLK                                                    FDB02510
0202              74 CONTINUE                                                         FDB02520
0203                DO 203 I=1,10                                                     FDB02530
0204                PX(I)=0                                                           FDB02540
0205                LN(I)=0                                                           FDB02550
0206             203 CONTINUE                                                         FDB02560
0207                GO TO 19                                                          FDB02570
0208             62 WRITE(6,68) (FLDNM(I),I=1,6)                                     FDB02580
0209             68 FORMAT(//5X,'ERROR-FOR FIELD ',6A1,' THE NUMBER OF PIXELS DOES NOTFDB02590
                1'/5X,'MATCH WITH THE NUMBER OF LINES')                              FDB02600
0210                GO TO 73                                                          FDB02610
                C*                                                                   FDB02620
                C*    FIELD                                                          FDB02630
                C*                                                                   FDB02640
0211              70 M=FINC(CARD2,COL,SBP)                                           FDB02650
0212                 IF (M.NE.1) GO TO 19                                            FDB02660
0213                 BCOL = COL                                                      FDB02670
0214                 LNV = 0                                                         FDB02680
0215                 PXV = 0                                                         FDB02690
0216                 DO 71 I=1,3                                                     FDB02700
0217                 IF(CARD2(COL-4).NE.CHNG(I)) GO TO 75                            FDB02710
0218                 GO TO (100,200,300),I                                           FDB02720
0219               75 CONTINUE                                                       FDB02730
0220               71 CONTINUE                                                       FDB02740
0221                 GO TO 70                                                        FDB02750
                C*                                                                   FDB02760
                C*    CLASS                                                          FDB02770
0222              100 DO 72 I=1,6                                                    FDB02780
0223                 CLNM(I) = BLK                                                   FDB02790
0224              72 CLNM(I)=NXTCHR(CARD2,COL)                                       FDB02800
0225                 WRITE (6,102) (CLNM(I),I=1,6)                                   FDB02810
0226                 PUNCH 101,(CLNM(I),I=1,6)                                       FDB02820
0227              101 FORMAT ('CLASSNAME',1X,6A1)                                    FDB02830
0228               102 FORMAT(5X,'CLASSNAME',1X,6A1)                                 FDB02840
0229                 GO TO 70                                                        FDB02850
                C*                                                                   FDB02860
                C*    LINE                                                           FDB02870
0230              200 LNC1=FIXNUM(CARD2(COL-2),MASK)                                 FDB02880
0231                 IF(LNC1.LT.0.OR.LNC1.GT.9)GO TO 4000                            FDB02890
0232                 LNC2=FIXNUM(CARD2(COL-1),MASK)                                  FDB02900
0233                 IF(LNC2.LT.0.OR.LNC2.GT.9) GO TO 4000                           FDB02910
0234                 LNCNT = LNC1*10 + LNC2                                          FDB02920
0235              202 M = FIND(CARD2,COL,BLK)                                        FDB02930
0236                 IF (M.NE.1) COL = 69                                            FDB02940
0237                 CDIF = COL-BCOL-1                                               FDB02950
0238                 DO 201 I=1,CDIF                                                 FDB02960
0239                 M = I-1                                                         FDB02970
0240                 W = COL - I                                                     FDB02980
0241                 LNVV = FIXNUM(CARD2(W),MASK)                                    FDB02990
0242                 IF(LNVV.LT.0.OR.LNVV.GT.9) GO TO 4000                           FDB03000
0243                 LNV = LNVV * (10**M) + LNV                                      FDB03010
0244              201 CONTINUE                                                       FDB03020
0245                 IF (PXFLAG.EQ.1) GO TO 301                                      FDB03030
0246                 IF (SMFLAG.EQ.1) GO TO 85                                       FDB03040
```

FILE FDBCVT

```
0247                IF (LNFLAG.EQ.1) GO TO 86                                   FDB03050
0248                LN(LNCNT) = LNV                                             FDB03060
0249                GO TO 70                                                    FDB03070
                C*                                                              FDB03080
                C*      PIXEL                                                   FDB03090
0250             300    PXC1=FIXNUM(CARD2(COL-2),MASK)                          FDB03100
0251                    IF(PXC1.LT.0.OR.PXC1.GT.9) GO TO 4000                   FDB03110
0252                    PXC2=FIXNUM(CARD2(COL-1),MASK)                          FDB03120
0253                    IF(PXC2.LT.0.OR.PXC2.GT.9) GO TO 4000.                  FDB03130
0254                    PXCNT = PXC1*10 + PXC2                                  FDB03140
0255                    PXFLAG = 1                                              FDB03150
0256                    GO TO 202                                               FDB03160
0257             301    PX(PXCNT) = LNV                                         FDB03170
0258                    PXFLAG = 0                                              FDB03180
0259                    GO TO 70                                                FDB03190
                C*                                                              FDB03200
                C*      BIAS                                                    FDB03210
                C*                                                              FDB03220
0260             90     BIKEY = 1                                               FDB03230
0261                    SAMPLE = 0                                              FDB03240
0262                    LINE = 0                                                FDB03250
0263              83    M = FIND(CARD2,COL,SEP)                                 FDB03260
0264                    IF(M.NE.1) GO TO 81                                     FDB03270
0265                    BCOL = COL                                              FDB03280
0266                    LNV = 0                                                 FDB03290
                C*                                                              FDB03300
                C*      FIND THE LINE(L) AND/OR SAMPLE(S) INCREMENT             FDB03310
                C*                                                              FDB03320
0267                    IF (CARD2(COL-1).EQ.CHNG(1)) GO TO 82                   FDB03330
0268                    IF(CARD2(COL-1).NE.SSS) GO TO 83                        FDB03340
0269                    SMFLAG = 1                                              FDB03350
0270                    GO TO 202                                               FDB03360
0271             85     SAMPLE = LNV                                            FDB03370
0272                    SMFLAG = 0                                              FDB03380
0273                    GO TO 83                                                FDB03390
0274             82     LNFLAG = 1                                              FDB03400
0275                    GO TO 202                                               FDB03410
0276             86     LINE=LNV                                                FDB03420
0277                    LNFLAG = 0                                              FDB03430
0278                    GO TO 83                                                FDB03440
0279               81   IF((LNFLAG.EQ.0).AND.(SMFLAG.EQ.0)) GO TO 19            FDB03450
0280                    WRITE(6,87) CODE,CARD2                                  FDB03460
0281             87     FORMAT(//5X,'ERROR IN BIAS CARD'/3X,A4,68A1/5X,'THE EQUALS SIGN IS FDB03470
                      1 MISSING FOR EITHER THE SAMPLE AND/OR LINE INCREMENT')   FDB03480
0282                    BIKEY = 0                                               FDB03490
0283                    GO TO 19                                                FDB03500
0284             4000   WRITE(6,4001)CODE,CARD2                                 FDB03510
0285             4001   FORMAT(5X,'***** CARD IN ERROR IS - ',A4,68A1)          FDB03520
0286                    GO TO 19                                                FDB03530
0287             5002   STOP                                                    FDB03540
0288                    END                                                     FDB03550
```

A-5
29

FILE FIND

```
0001                INTEGER FUNCTION FIND(CARD,COL,VECTOR)                    FIN00010
0002                IMPLICIT INTEGER (A-Z)                                    FIN00020
0003                DIMENSION CARD(68)                                        FIN00030
0004                DATA CRDSIZ/68/,BLANK/' '/,COMMA/','/                     FIN00040
                C*                                                            FIN00050
                C*   THE FIND SUBROUTINE SEARCHES FROM CARD(COL+1) TO CARD(CRDSIZ)  FIN00060
                C*   FOR THE CHARACTER(S) IN VECTOR                           FIN00070
                C*                                                            FIN00080
0005                L = COL + 1                                               FIN00090
0006                IF(L.GT.CRDSIZ) GO TO 15                                  FIN00100
0007                I=1                                                       FIN00110
0008                DO 10 K=L,CRDSIZ                                          FIN00120
0009                COL = K                                                   FIN00130
0010                IF (CARD(COL).EQ.VECTOR) GO TO 20                         FIN00140
0011             10 CONTINUE                                                  FIN00150
0012             15 I=-1                                                      FIN00160
0013                COL = L-1                                                 FIN00170
0014             20 CONTINUE                                                  FIN00180
0015                FIND = I                                                  FIN00190
0016                RETURN                                                    FIN00200
0017                END                                                       FIN00210
```

A-6
30

FILE NXTCHR

```
0001                FUNCTION NXTCHR(CARD,COL)                                   NXT00010
0002                IMPLICIT INTEGER (A-Z)                                      NXT00020
0003                DIMENSION CARD(68)                                          NXT00030
0004                DATA CRDSIZ/68/,BLANK/' '/,COMMA/','/                       NXT00040
                C*                                                              NXT00050
                C*   THE NXTCHR SUBROUTINE SEARCHES FROM CARD(COL+1) TO         NXT00060
                C*   CARD(CRDSIX) FOR THE NEXT NONBLANK CHARACTER               NXT00070
                C*                                                              NXT00080
0005                L=COL + 1                                                   NXT00090
0006                IF (L.GT.CRDSIZ) GO TO 40                                   NXT00100
0007                DO 30 COL=L,CRDSIZ                                          NXT00110
0008                NXTCHR = CARD(COL)                                          NXT00120
0009                IF(NXTCHR.NE.BLANK)GO TO 50                                 NXT00130
0010             30 CONTINUE                                                    NXT00140
0011                COL = CRDSIZ - 1                                            NXT00150
0012             40 NXTCHR = BLANK                                              NXT00160
0013             50 CONTINUE                                                    NXT00170
0014                RETURN                                                      NXT00180
0015                END                                                         NXT00190
```

31

FILE FIXNUM

```
0001                  INTEGER FUNCTION FIXNUM(NUM,MASK)                                  FIX00010
                C*                                                                       FIX00020
                C*    FIXNUM SUBROUTINE WILL TAKE NUM ARGUMENT (WHICH IS                 FIX00030
                C*    IN CHARACTER FORMAT) AND CONVERT IT INTO AN                        FIX00040
                C*    INTEGER.                                                           FIX00050
                C*                                                                       FIX00060
                C*                                                                       FIX00070
                      IMPLICIT INTEGER (A-Z)                                             FIX00080
0002                  NUM1 = NUM * (2.0 **(-24.0))                                       FIX00090
0003                  NUM1 = NUM1- MASK                                                  FIX00100
0004                  IF ( (NUM1.GE.0).AND.(NUM1.LE.9))GO TO 100                         FIX00110
0005                  WRITE(6,90)                                                        FIX00120
0006                90 FORMAT(/5X,'**ERROR - NUMERIC CHARACTER EXPECTED AND NOT FOUND') FIX00130
0007               100 FIXNUM = NUM1                                                     FIX00140
0008                   RETURN                                                          FIX00150
0009                   END
0010
```

A-8
32

APPENDIX B

PROGRAM VERIFICATION INPUT AND OUTPUT

```
 L JIC200
ENTER PASSWORD:
****
ENTER NAME:GROP
YOUR OPERATORS THIS AFTERNOON ARE  DEANIE  AND  DOUG.
ANY POSSIBLE SHUTDOWN IS 01(9-0900 MED.
READY AT 14.58.00 ON 06/21/77
CP
#I CMS
CMS (VER 3.2) READY;

#DISK SET S
LINE END CHARACTER SET TO ^
YOU ARE LINKED TO TEMP DISK 21
P (192): 11 FILES; 206 REC IN USE, 90 LEFT (OF 296), 70% FULL (2 CYL)
P; T=1.90/2.84 14.58.34

>
** CARDS  XFERED BY HOUSTON **
CMS
>D READ FDBCVT DATA
R; T=0.08/0.25 14.59.36

:FDBCVT
14.59.44 FILEDEF 5 DSK-P1 FDBCVT DATA
14.59.46 FILEDEF 6 DSK-P1 PRINT LISTING
14.59.48 FILEDEF 7 DSK-P1 PUNCH OUTPUT
14.59.50 LOAD FDBCVT (XEQ)
EXECUTION BEGINS...
P; T=1.69/2.42 15.00.09

>D PRINT PRINT LISTING
R; T=0.18/0.48 15.00.22

>D PUNCH PUNCH OUTPUT
R; T=0.04/0.11 15.01.42

:LOGOUT
CONNECT= 00:04:51  VIRTCPU= 000:03.91  TOTCPU= 000:06.47
LOGOUT AT 15.02.47 ON 06/21/77
P-67 ONLINE
```

```
FILE . . .     ERRCVT   DATA     P1                                          PAGE 001
SEGSTART ID=I1234
FLDSTART NAME=LINGL
FIELD LINE01=20 PIXEL01=30 LINE02=30 PIXEL02=10 LINE03=80 PIXEL03=40
FLDEND
FLDSTART NAME=LINTET
FIELD LINE01=30 PIXEL01=30 LINE02=30 PIXEL02=50 LINE03=80 PIXEL03=40
FLDEND
FLDSTART NAME=SAND
FIELD LINE01=100 PIXEL01=20 LINE02=80 PIXEL02=50
FIELD LINE03=50 PIXEL03=50 LINE04=60 PIXEL04=20
FIELD LINE05=50 PIXEL05=10 LINE06=85 PIXEL06=10
FLDEND
FLDSTART NAME=RYE
FIELD PIXEL01=75 LINE01=20 PIXEL02=90 LINE02=30
FIELD PIXEL03=95 LINE03=25 PIXEL04=90 LINE04=10
FLDEND
FLDSTART NAME=WHEAT
FIELD CLASS=W*0101 TYPE=T
FIELD PIXEL01=10 LINE01=10 PIXEL02=20 LINE02=1
FIELD LINE03=30 PIXEL03=20 LINE04=30 PIXEL04=10
FLDEND
BIAS S=2 L=3
FLDSTART NAME=BARLEY
FIELD LINE01=1 PIXEL01=15 LINE02=3 PIXEL02=10
FIELD LINE03=5 PIXEL03=12 LINE04=10 PIXEL04=8
FIELD LINE05=15 PIXEL05=13 LINE06=11 PIXEL06=15
FIELD LINE07=7 PIXEL07=13
FLDEND
FLDSTART NAME=NEWWHT
FIELD TYPE=T CLASS=W*0120
FIELD PIXEL01=20 LINE01=20 PIXEL02=30 LINE02=20
FIELD PIXEL03=15 LINE03=40
FLDEND
SEGEND
SEGSTART ID=I5678
FLDSTART NAME=WHEAT
FIELD CLASS=W*0102
FIELD LINE01=2 LINE02=14
FIELD PIXEL01=7 PIXEL02=9
FLDEND
SEGEND
SEGSTART ID
FLDSTART NAME=WHEAT
FIELD LINE01=10 PIXEL01=10 LINE02=20 PIXEL02=20
FLDEND
SEGEND
SEGSTART ID=I7890
FLDSTART NAME
FIELD LINE01=1 PIXEL01=2 LINE02=3 PIXEL02=4
FLDEND
FLDSTART NAME=FALLOW
FIELD  PIXEL01=50 LINE01=50 PIXEL02=10 LINE02=50 PIXEL03=20 LINE03=20
FLDEND
SEGSTART ID=I6789
BIAS L=3
FLDSTART NAME=GRASS
FIELD LINE01=1 LINE02=10 PIXEL01=1 PIXEL02=10
FLDSTART NAME=HAY
FIELD LINE01=1 LINE02=10 LINE03=30
FIELD PIXEL01=1 PIXEL02=15
FLDEND
FIELD LINE01=100 PIXEL01=100
FIELD NAME=BOMB
FLDEND
SEGEND
SEGSTART ID=I9673
FLDSTART NAME=ALL
FIELD  TYPE=T CLASS=W*WW01
FIELD LINE01=01 PIXEL01=01 LINE02=01 PIXEL02=196 LINE03=117 PIXEL03=196
FIELD LINE04=117 PIXEL04=01
FLDEND
DEFLT CAT=N AP=50 TH=1.0
SEGEND
```

B-2
35

```
   LINGI       %1,1¤,% 30, 30¤,% 40, 80¤,% 10, 30¤

   LINTFT      %1,1¤,% 30, 30¤,% 50, 30¤,% 40, 80¤

   SAND        %1,1¤,% 20,100¤,% 10, 85¤,% 10, 50¤,% 20, 60¤
         ,% 50, 50¤,% 50, 80¤

   RYE         %1,1¤,% 75, 20¤,% 90, 10¤,% 95, 25¤,% 40, 30¤
   CLASSNAME W=0101

   WHEAT       %1,1¤,% 10, 10¤,% 20,  1¤,% 20, 30¤,% 10, 30¤

   BARLEY      %1,1¤,% 17,  4¤,% 15, 10¤,% 17, 14¤,% 15, 18¤
         ,% 10, 13¤,% 14,  8¤,% 12,  6¤

   NONWHT      %1,1¤,% 22, 23¤,% 32, 23¤,% 17, 43¤
   COMMENT       SAMPLE SEGMENT  15678
   CLASSNAME W=0102

   WHEAT       %1,1¤,%  7,  2¤,%  9, 14¤

   ERROR--THE SEGSTART CARD
SEGSTART ID
   IS MISSING  AN EQUALS SIGN--LOOK FOR THE NEXT SEGSTART OR EOF
   COMMENT       SAMPLE SEGMENT  17890

   ERROR--A VALID SEGSTART CARD BEFORE
SEGSTART ID=17890
   IS MISSING

   ERROR--THE FLDSTART CARD
FLDSTART NAME
   IS MISSING  AN EQUALS SIGN--LOOK FOR THE NEXT FLDSTART OR SEGEND CARD

   ERROR--A VALID FLDSTART CARD BEFORE
FLDSTART NAME=FALLOW
   IS MISSING

   FALLOW      %1,1¤,% 50, 50¤,% 10, 50¤,% 20, 20¤
   COMMENT       SAMPLE SEGMENT  16789

   ERROR---A SEGEND CARD BEFORE
SEGSTART ID=16789
   IS MISSING

   ERROR--A FLDEND CARD BEFORE
FLDSTART NAME=HEY
   IS MISSING

   ERROR-FOR FIELD HEY    THE NUMBER OF PIXELS DOES NOT
   MATCH WITH THE NUMBER OF LINES

   %%ERROR - NUMERIC CHARACTER EXPECTED AND NOT FOUND
   %%%% CARD IN ERROR IS - FIELD NAME=BOMB

               %1,1¤,%100,103¤
   COMMENT       SAMPLE SEGMENT  19673

   ERROR---A SEGEND CARD BEFORE
SEGSTART ID=19673
   IS MISSING

   ERROR--A VALID FLDSTART CARD BEFORE
FLDSTART NAME=ALL
   IS MISSING
   CLASSNAME W=0101
```