U.S. DEPARTMENT OF COMMERCE
NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
NATIONAL WEATHER SERVICE
NATIONAL METEOROLOGICAL CENTER

OFFICE NOTE 184

Procedures for Creating or Unpacking FGGE Level III Data Sets in International Exchange Formats

Armand J. Desmarais
Development Division

JULY 1978

This is an unreviewed manuscript, primarily intended for informal exchange of information among NMC staff members.

Procedures for Creating or Unpacking FGGE Level III Data Sets

General

The documentation for FGGE Level III formats is described in Appendix 11, FGGE International Data Management Plan, "Format for the International Exchange of Level III Data Sets During the FGGE."

Purpose

This note describes the basic steps needed for creating or unpacking Level III data sets with the use of attached subroutine listings. These subroutines were written for use on IBM 360 machines that have 32-bit words. Some modifications may be necessary to utilize these routines on other machines.

Level III Tape Files

File 1 -- TEST file

File 2 -- TAPE HEADER file

File 3 -- GRID DESCRIPTOR file

Files 4-n -- LEVEL III DATA file(s)

To Create Level III Magnetic Tape

The basic sequence of events in a computer program should be as follows, assuming that the required analysis fields are available in an analysis file and are ready for processing:

A. Write the required TEST file on output tape (File 1).

B. Screen available analyses to determine first and last dates to be processed and written on the output tape. Prepare the documentation for the TAPE HEADER file with appropriate dates (File 2), and write it on the output tape (See example, Attachment 1).

C. Describe the grid(s) used for the Level III data (See example, Attachment 1). Write GRID DESCRIPTOR file on output tape.

D. Select an analysis field from the analysis file, prepare the necessary unique identification words (see subroutine W3FI32, Attachment 2), scale and pack the data (see subroutine W3AI00, Attachment 3), and write the packed field in a LEVEL III DATA file on the output tape.

E. Repeat D above for other fields for the given date/time.

F. Write an end-of-file tape mark on the output tape.

G. Repeat steps D, E, and F above for any remaining date/times.

H. Terminate the output tape with at least two (2) consecutive end-of-file tape marks.

To Unpack Level III Data

The format required to read FGGE Level III data sets is written in clear text in the second file (TAPE HEADER) of each tape. The third file of each tape will contain information concerning the arrangement of the data for the grid(s) used in the DATA file(s) that start with the fourth file.

A. Read a record from selected DATA file.

B. Determine if you desire to process and unpack the record. Subroutine W3FI33 (Attachment 4) could be used to convert the first 256 bits of the record identification to individual field identifiers.

C. If the field is desired, the packed data (beginning at bit 385) can be unpacked and rescaled with the use of subroutine W3AI01 (see Attachment 5). If not desired, repeat A and B above.

A sample FORTRAN program to locate and unpack N. Hemisphere 700 mb heights, and to print out some values over the United States, is given in Attachment 6. Descriptions of built-in and intrinsic functions used in the various subroutines are given in Attachment 7.

EXAMPLE -- TAPE HEADER FILE

```
FGGE3A003178010200780108l2
 10780          FIXED LENGTH RECORDS (BLOCKSIZE = RECORD SIZE)
 20A4           USE  ( 100(27A4) )  FOR FULL RECORD
 0123456789=:>  /STUVWXYZ.(-JKLMNOPQR*);+ABCDEFGH.)(<
 9-TRACK,  800 BPI
 BINARY,  ODD PARITY
 IBM 360/195,       32 BITS/WORD,  8 BITS/BYTE.
 NATIONAL METEOROLOGICAL CENTER, NWS, NOAA, WASHINGTON, D.C., USA
  ( 100(27A4) )
  HOUGH FUNCTION ANALYSIS METHOD, GLOBAL      (FLATTERY)
    FIRST GUESS COEFFICIENTS DERIVED FROM A 9-LAYER PRIMITIVE EQUATION
    FORECAST MODEL ON A 2.5 X 2.5 LATITUDE/LONGITUDE GRID.
  100. =  (42640000) HEXADECIMAL            FLOATING POINT REPRESENTATION
  -100. = (C2640000) HEXADECIMAL            FLOATING POINT REPRESENTATION

    ANALYSIS FIELDS NORMALLY PROVIDED:
     12 LEVELS OF U- and V-WIND COMPONENTS, TEMPERATURES, AND HEIGHTS
       (AT 1000, 850, 700, 500, 400, 300, 250, 200, 150, 100, 70, AND 50 MB);
      6 LEVELS OF RELATIVE HUMIDITIES (AT 1000, 850, 700, 500, 400, AND 300 MB);
       TROPOPAUSE PRESSURE AND TEMPERATURE (MODELED);  SNOW FIELD;
       SEA SURFACE TEMPERATURE (FROM SATELLITE DATA); SEA LEVEL PRESSURE.

MISSING ANALYSES:     NONE

  ...   END OF TEXT  ........................................
```

EXAMPLE -- GRID DESCRIPTOR FILE

```
GRID 029          RECTANGLE
GEOGRAPHIC
RIGHT-HANDED
GRIDLINES = 037, DELTAJ = 2.50, FIRSTPT=(0.00N, 0.00E)
J=ALL, NI=145,   DELTAI = 2.50
 ...  END OF TEXT  ..................................


GRID 030          RECTANGLE
GEOGRAPHIC
RIGHT-HANDED
GRIDLINES = 037, DELTAJ = 2.50, FIRSTPT=( 90.00S, 0.00E)
J=ALL, NI=145,   DELTAI = 2.50
 ...  END OF TEXT  ..................................
```

W3FI32
Pack ID
(Modified for FGGE Level III)

FORTRAN H EXT +
IBM 360
NMC, DCA

PURPOSE:

To convert an array of the 27 data field identifiers into an array of the first 8 identification words. (See Appendix 11, FGGE International Data Management Plan, "Format for the International Exchange of Level III Data Sets During the FGGE."

USAGE:

Calling Statement

```
CALL W3FI32(KARRAY,PKDNT)
```

where,

KARRAY = array containing the 27 data field identifiers (INTEGER*4).
PKDNT = array to receive the 8 identification words.

NOTE 1

(a) If any number n in (KARRAY(I),I=1,27) is erroneously large, W3FI32 will print:
'VALUE IN KARRAY(I)=n ID TOO LARGE TO PACK'
(b) If any number n in KARRAY(I) is erroneously negative, W3FI32 will print:
'VALUE IN KARRAY(I)=n SHOULD NOT BE NEGATIVE'
(c) If either (a) or (b) occurs, that portion of the packed word corresponding to KARRAY(I) will be set to binary ones.

Examples

1. Suppose KARRAY(4)=30. Then W3FI32 will print 'VALUE IN KARRAY(4)=30 IS TOO LARGE TO PACK' and will place ones in the first 4 bits of PKDNT(2).

2. To form PKDNT from the 27 data field identifers for 500 mb height from FMANL (limited-area fine mesh analysis file) dated 00Z March 15, 1974, KARRAY should be initialized as follows:

KARRAY	IDENTIFIER	KARRAY	IDENTIFIER	KARRAY	IDENTIFIER
(1)=1	Q	(10)=0	F_2	(19)=0	Unused
(2)=8	S_1	(11)=0	N	(20)=1523	NW
(3)=0	F_1	(12)=0	C_2	(21)=74	JJ
(4)=0	t_1	(13)=0	E_2	(22)=3	MM
(5)=50000	C_1	(14)=0	CD	(23)=15	YY
(6)=-2	E_1	(15)=0	CM	(24)=0	GG
(7)=0	m	(16)=0	KS	(25)=0	R
(8)=0	X	(17)=5	K	(26)=19	G
(9)=0	S_2	(18)=0	Unused	(27)=3021	J

Then the resulting PKDNT array would be as follows (in hexadecimal):

PKDNT	(1)	00100800	(5)	00000005
	(2)	00C35082	(6)	000005F3
	(3)	00000000	(7)	4A030F00
	(4)	00000000	(8)	00130BCD

W3FI32

```
      SUBROUTINE W3FI32 (IARRAY,KIDNT)
C     SUBROUTINE W3FI32 PACKS DATA FIELD IDENTIFIERS.
      DIMENSION KIDNT(8),ITABLE(27),IARRAY(27)
      DATA ITABLE/Z00140C01,Z00080C01,Z00000801,Z001C0402,Z01081402,
     1Z01000802,Z001C0403,Z00140803,Z00080C03,Z00000803,Z001C0404,
     2Z01081404,Z01000804,Z00180805,Z00100805,Z00080805,Z00000805,
     3Z001C0406,Z00100C06,Z00001006,Z00180807,Z00100807,Z00080807,
     4Z00000807,Z00180808,Z00100808,Z00001008/
      DATA KX/ZFFFFFFFF/
      DATA MASK/Z000000FF/
C     MAKE KIDENTS=0
      DO 20 I=1,8
      KIDNT(I)=0
   20 CONTINUE
      ISIGN=0
      DO 40 I=1,27
      ISC=ITABLE(I)
      I1=LAND(ISC,MASK)
      I2=LAND( SHFTR(ISC,8),MASK)
      I3=LAND( SHFTR(ISC,16),MASK)
      I4=LAND(SHFTR(ISC,24),MASK)
C     SIGN TEST
      IV=IARRAY(I)
      IF(IV.GE.0)GO TO 12
      IF(I4.NE.0) GO TO 1
      WRITE (6,22) I,IV
   22 FORMAT(///,1X,' VALUE IN KARRAY(',I2,')=',I11,' SHOULD NOT BE NEGA
     /TIVE'///)
      GO TO 30
    1 IV=IABS(IV)
      MSIGN=1
      ISIGN=MSIGN
      K=I2/4
      DO 25 M=1,K
   25 ISIGN=SHFTL(ISIGN,4)
      ISIGN=SHFTR(ISIGN,1)
      IV=LOR(IV,ISIGN)
   12 CONTINUE
C     MAG TEST
      IF( SHFTR(IV,I2).EQ.0)GO TO 100
      WRITE(6,21)I,IV
   21 FORMAT(///,1X,' VALUE IN KARRAY(',I2,')=',I11,' IS TOO LARGE TO PA
     /CK'///)
   30 IV=KX
      IA=32-I2
      IV=SHFTR(IV,IA)
C     SHIFT
  100 KIDNT(I1)=LOR(KIDNT(I1), SHFTL(IV,I3))
   40 CONTINUE
      RETURN
      END
```

W3AI00

```
      SUBROUTINE W3AI00(REAL4,PACK,LABEL)
C
CC SCALE AND CONVERT A FIELD OF REAL*4 DATA TO HALF-WORD INTEGERS AND
C  PACK TOGETHER WITH LABEL DATA IN FGGE LEVEL III DATA FORMAT.
CC
C     CALL W3AI00(REAL4,PACK,LABEL)
C
C  WHERE REAL4 = ARRAY OF REAL*4 DATA TO BE PACKED.
C        PACK  = OUTPUT ARRAY. WORDS 1-8 COPY LABEL WORDS 1-8, WORDS
C                9-12 ARE GENERATED, WORDS 13,14,... ARE PACKED DATA.
C                ALLOW 12+(J+1)/2 FULLWORDS, WHERE J IS NUMBER OF POINTS.
C        LABEL =INPUT LABEL DATA. WORD 8, BITS 16-31, MUST CONTAIN THE
C                COUNT OF THE NUMBER OF POINTS IN ARRAY REAL4 FOR THIS
C                ROUTINE TO WORK.
C
C  GENERATED DATA TO BE FOUND IN PACK.....
C  WORD 9  BITS  0-15: NUMBER OF BYTES IN WHOLE RECORD.
C          BITS 16-31: EXCLUSIVE-OR CHECKSUM OF WHOLE RECORD (EXCEPT
C                      CHECKSUM ITSELF), BY HALFWORDS.
C  WORD 10 BITS  0-31: REAL*4 CENTERING VALUE A = THE MEAN OF THE MAX
C                      AND MIN VALUES OF ARRAY REAL4.
C  WORD 11 BITS  0-23: ZERO
C          BITS 24-31: INTEGER*2 SHIFT VALUE N, THE LEAST INTEGER SUCH
C                      THAT ABS(X-A)/2**N IS LESS THAN 1 FOR ALL X IN
C                      ARRAY REAL4.  N IF NEGATIVE IS IN 2'S COMPLEMENT
C                      FORM.  VALUE OF N WILL NOT EXCEED 127.
C  WORD 12 BITS  0-31: ZERO
C  WORD 13 BITS  0-15: FIRST PACKED DATUM
C          BITS 16-31: SECOND PACKED DATUM
C  WORD 14 BITS  0-15: THIRD PACKED DATUM
C    . . .
C
C  THE BINARY POINT OF THE PACKED DATA IS CONSIDERED TO BE TO THE RIGHT
C  OF THE LEFTMOST, OR SIGN BIT.  DATA IS IN 2'S COMPLEMENT FORM IF
C  NEGATIVE.  IF PACK IS EQUIVALENCED TO AN INTEGER*2 ARRAY HLF, THE
C  DATA MAY BE RECONSTRUCTED AS FOLLOWS:
C       REAL4(J) = HLF(J+24)*2.**(N-15) + A
C  WHERE A AND N ARE TO BE FOUND IN WORDS 10 AND 11.
C
      REAL*4 REAL4(1)
      INTEGER*2 LABEL(1),PACK(1),IA(2)
      EQUIVALENCE (A,IA(1)),(X,IX)
C
CC TRANSFER LABEL DATA TO WORDS 1-8.  GET WORDCOUNT, COMPUTE BYTES.
C
      DO 10 I=1,16
      PACK(I) = LABEL(I)
 10   CONTINUE
      J = LABEL(16)
      M = J+24
      PACK(17) = M+M
      PACK(18) = 0
C
CC FIND MAX, MIN OF DATA, COMPUTE A AND N.
C
      RMAX = REAL4(1)
      RMIN = RMAX
      DO 20 I=2,J
```

```
      RMAX = AMAX1(RMAX,REAL4(I))
      RMIN = AMIN1(RMIN,REAL4(I))
   20 CONTINUE
      A = 0.5*(RMAX+RMIN)
      X = RMAX-A
      N = LAND(SHFTR(IX,24),127)
      N = 4*(N-64)
      IF (TBIT(IX,8)) GO TO 30
      N = N-1
      IF (TBIT(IX,9)) GO TO 30
      N = N-1
      IF (TBIT(IX,10)) GO TO 30
      N = N-1
   30 N = MAX0(-127,MIN0(127,N))
      PACK(19) = IA(1)
      PACK(20) = IA(2)
      PACK(21) = 0
      PACK(22) = N
      PACK (23) = 0
      PACK(24) = 0
C
CC NOW PACK UP THE DATA
C
      TWON = 2.**(15-N)
      DO 60 I=1,J
      X = (REAL4(I)-A)*TWON
      K = X+SIGN(0.5,X)
      IF (K.LT.(-32767)) GO TO 40
      K = MIN0(32767,K)
      GO TO 50
   40 K = -32767
   50 PACK(I+24) = K
   60 CONTINUE
C
CC    COMPUTE CHECKSUM AND STORE
C
      IXOR = 0
      DO 70 I=1,M
      K = PACK(I)
      IXOR = LXOR(IXOR,K)
   70 CONTINUE
      PACK(18) = IXOR
      RETURN
      END
```

W3FI33
Unpack ID
(Modified for FGGE Level III)

FORTRAN H EXT +
IBM 360
NMC,DCA

PURPOSE:

To convert an array of the first 8 identification words into an array of 27 data field identifiers. (See Appendix 11, FGGE International Data Management Plan, "Format for the International Exchange of Level III Data Sets During the FGGE.")

Calling Statement

```
CALL W3FI33(PKDNT,KARRAY)
```

where,

PKDNT = array containing the 8 identification words.
KARRAY = array to receive the 27 data field identifiers (INTEGER*4).

Example

Suppose the 8 identification words for 500 mb height from FMANL (limited-area fine mesh analysis file) dated 00Z March 15, 1974 are given (in hexadecimal)

PKDNT	(1)	00100800	(5)	00000005
	(2)	00C35082	(6)	000005F3
	(3)	00000000	(7)	4A030F00
	(4)	00000000	(8)	00130BCD

Then the resulting KARRAY array would be as follows:

KARRAY	IDENTIFIER	KARRAY	IDENTIFIER	KARRAY	IDENTIFIER
(1)=1	Q	(10)=0	F_2	(19)=0	Unused
(2)=8	S_1	(11)=0	N^2	(20)=1523	NW
(3)=0	F_1	(12)=0	C_2	(21)=74	JJ
(4)=0	t^1	(13)=0	E_2	(22)=3	MM
(5)=50000	C_1	(14)=0	CD	(23)=15	YY
(6)=-2	E_1	(15)=0	CM	(24)=0	GG
(7)=0	m^1	(16)=0	Ks	(25)=0	R
(8)=0	X	(17)=5	K	(26)=19	G
(9)=0	S_2	(18)=0	Unused	(27)=3021	J

W3FI33

```
      SUBROUTINE W3FI33 (IDNT,LARRAY)
      DIMENSION LARRAY(27),IDNT(8),ITABLE(27),J(6),JS(6)
C     SUBROUTINE W3FI33 UNPACKS THE 8 IDENTIFICATION WORDS INTO AN ARRAY
C     OF 27 DATA FIELD IDENTIFIERS.
      DATA J/Z00000000F,Z000000FF,Z00000FFF,Z0000FFFF,Z000FFFFF,
     /Z00FFFFFF/
      DATA JS/Z00000007,Z0000007F,Z000007FF,Z00007FFF,Z0007FFFF,
     /Z007FFFFF/
      DATA MASK/Z000000FF/
      DATA ITABLE/Z00140C01,Z00080C01,Z00000801,Z001C0402,Z01081402,
     1Z01000802,Z001C0403,Z00140803,Z00080C03,Z00000803,Z001C0404,
     2Z01081404,Z01000804,Z00180805,Z00100805,Z00080805,Z00000805,
     3Z001C0406,Z00100C06,Z00001006,Z00180807,Z00100807,Z00080807,
     4Z00000807,Z00180808,Z00100808,Z00001008/
      DO 50 I=1,27
      ISC=ITABLE(I)
      I1=LAND(ISC,MASK)
      I2=LAND(SHFTR(ISC,8),MASK)
      I3=LAND( SHFTR(ISC,16),MASK)
      I4=LAND(SHFTR(ISC,24),MASK)
      IX=I2/4
      LARRAY(I)=LAND(SHFTR(IDNT(I1),I3),J(IX))
C     TO SEE IS THE NUMBER IS MINUS
      IF(I4.EQ.0) GO TO 50
      IB=LARRAY(I)
      IC=SHFTL(IB,1)
      DO 30 M=1,IX
   30 IC=SHFTR(IC,4)
      IF(IC.EQ.0) GO TO 50
C     ABS. VALUE,THEN MINUS.
      LARRAY(I)=-(LAND(IB,JS(IX)))
   50 CONTINUE
      RETURN
      END
```

W3AI01

```
      SUBROUTINE W3AI01(PACK,REAL4,LABEL)
C
C UNPACK AND FLOAT A FIELD OF PACKED DATA IN FGGE LEVEL III DATA
C   FORMAT ACCORDING TO SPECIFICATIONS IN THE ID WORDS OF THE
C PACKED DATA.  ALSO MOVE THE ID WORDS TO A LABEL ARRAY.
C
C      CALL W3AI01(PACK,REAL4,LABEL)
C
C WHERE PACK  = ARRAY OF ID WORDS AND PACKED DATA TO BE UNPACKED.
C       REAL4 = REAL*4 ARRAY TO RECEIVE THE UNPACKED DATA WHICH WILL
C               BE DE-SCALED ACCORDING TO ID SPECIFICATIONS.
C       LABEL = A 12-WORD ARRAY INTO WHICH THE ID WORDS WILL BE COPIED.
C
C PACK MUST CONTAIN THE FOLLOWING FIELDS.....
C WORD  8 BITS 16-31: NUMBER OF DATA POINTS J.
C WORD  9 BITS 16-31: EXCLUSIVE-OR CHECKSUM OF REST OF ARRAY BY
C                     HALFWORDS.  THUS CHECKSUM OF ENTIRE ARRAY SHOULD
C                     BE ZERO.
C WORD 10 BITS  0-31: REAL*4 DATA OFFSET VALUE A.
C WORD 11 BITS 16-31: INTEGER*2 SHIFT VALUE N, IN 2'S COMPLEMENT FORM
C                     IF NEGATIVE.
C WORD 13 BITS  0-15: DATUM 1
C      BITS 16-31: DATUM 2
C WORD 14 BITS  0-15: DATUM 3
C      ---     ETC   ----
C
C THERE WILL BE 12+(J+1)/2 FULLWORDS IN ARRAY PACK, AND J FULLWORDS IN
C ARRAY REAL4.
C
      REAL*4 REAL4(1)
      INTEGER*2 LABEL(1),PACK(1),IA(2)
      EQUIVALENCE (A,IA(1))
C
C TRANSFER ID WORDS TO LABEL.
C
      DO 10 I=1,24
      LABEL(I) = PACK(I)
 10   CONTINUE
C
C GET WORD COUNT, A, N.
C
      J = PACK(16)
      IA(1) = PACK(19)
      IA(2) = PACK(20)
      N = PACK(22)
      TWON = 2.**(N-15)
C
C UNPACK, CONVERT REAL*4 DATA
C
      DO 20 IM=1,J
      I = J+1-IM
      REAL4(I) = PACK(I+24)*TWON+A
 20   CONTINUE
      RETURN
      END
```

```
REQUESTED OPTIONS: MAP,XREF,LC(80)

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(80) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT MAP NOFORMAT NOGOSTMT XREF ALC NOANSF NOTERM FLAG(I)

                       FUNCTIONS INLINE ARE:   NONE


                C***    SAMPLE PROGRAM TO READ AND UNPACK FGGE LEVEL III 2.5 X 2.5
                C***   LATITUDE/LONGITUDE ANALYSIS FIELDS,  WRITTEN IN INTERNATIONAL
                C***   EXCHANGE FORMAT.
                C***   THIS SAMPLE SEARCHES FOR AND UNPACKS 700 MB HEIGHTS (NO. HEMIS.)
                C***    AND DISPLAYS VALUES OVER THE UNITED STATES --- FROM 30N TO 50N,
                C***      AND FROM 235E(125W) TO 280E(80W).
                C                  --- DESMARAIS, NMC, WASHINGTON.
  ISN 0002            DIMENSION LABEL(12), IBUF(2695), RECT(145,37), IDS(5)
  ISN 0003            LOGICAL*1   LABX(48)
  ISN 0004            DATA IDS / Z00100800, Z01117082, 0, 0,  Z11D /
  ISN 0005            EQUIVALENCE (LABX(1),LABEL(1))
  ISN 0006            NFILES = 1
                C***   ASSUME THAT THE TAPE  IS PRE-POSITIONED.....................
  ISN 0007            DO  500  KK = 1, NFILES
  ISN 0008            N = 0
  ISN 0009          7 N = N + 1
  ISN 0010            READ (10, 3030, ERR=307, END=320)  IBUF
                C***   SEARCH FOR DESIRED FIELD.
  ISN 0011       3030 FORMAT ( 255 (33A4) )
  ISN 0012            DO  12  I = 1, 5
  ISN 0013            IF (IBUF(I) .NE. IDS(I))  GO TO 7
  ISN 0014         12 CONTINUE
                C***  FOUND DESIRED FIELD          (700 MB HEIGHTS)
  ISN 0016            PRINT 13, (IBUF(I), I=1,12)
  ISN 0017         13 FORMAT ('    FOUND FIELD WITH THE FOLLOWING IDENTS...',/ 1X,12Z9,
                     1////)
                C***   UNPACK FROM IBUF  TO RECT
  ISN 0018            CALL W3AI01 (IBUF, RECT, LABEL)
                C
                C***      RECT(1,1)    AT 0 DEGREES N, 0 DEGREES EAST
                C***      RECT(2,1)    AT 0 DEGREES N, 2.5 DEGREES E.
                C***      RECT(3,1)    AT 0 DEGREES N, 5.0 DEGREES E.      ETC.
                C***      RECT(145,1) AT 0 DEGREES N,   0 DEGREES E.
                C***      RECT(1,2)     AT 2.5 DEGREES N, 0 DEGREES E.
                C***      RECT(2,2)     AT 2.5 DEGREES N, 2.5 DEGREES E.   ETC
                C     RECT(1,37) THRU (RECT(145,37) AT NORTH POLE.
                C
                C***  FOR SOUTHERN HEMISPHERE GRIDS   (5TH IDENTIFICATION WORD = 1E,)
                C     RECT(1,1) THRU RECT(145,1) WOULD BE AT SOUTH POLE.
                C***   RECT(1,37) AT 0 DEGREES N, 0 DEGREES E.
                C***  RECT(2,37) AT 0 DEGREES N, 2.5 DEGREES E.       ETC
                C
                C***  PRINT SELECTED 700 MB HEIGHTS (METERS)
  ISN 0019            PRINT 21
  ISN 0020         21 FORMAT ('      SELECTED 700 MB HEIGHTS (METERS) OVER THE UNITED ST
                     1ATES. (LOWER LEFT VALUE AT 30N, 125W)',///)
  ISN 0021            JJ = 21
  ISN 0022            DO  40  J = 1, 9
  ISN 0023            PRINT 22, (RECT(I,JJ), I=95, 113)
  ISN 0024         22 FORMAT (1X, 19F6.0,///)
  ISN 0025            JJ = JJ - 1
  ISN 0026         40 CONTINUE
  ISN 0027            PRINT 45, LABX(4), (LABX(I),I=25,28)
  ISN 0028         45 FORMAT (///,'   VALID ',I2,' HOURS AFTER ',I2,'/',I2,'/',I2,
                     1', ', I2,'00 GMT.')
  ISN 0029            PRINT 310
  ISN 0030        310 FORMAT (///,'   .....  RAN TO PLANNED EXIT ..........')
  ISN 0031            STOP
  ISN 0032        307 PRINT 308, KK, N
  ISN 0033        308 FORMAT (' READ ERROR ON UNIT 10, FILE NUMBER ',I2,', RECORD ',
                     1  I3,'.     WILL CONTINUE...')
  ISN 0034            GO TO 7
  ISN 0035        320 PRINT 321, KK
  ISN 0036        321 FORMAT (//,' --- HIT END-OF-FILE MARK NUMBER ',I2,' ON UNIT 10.')
  ISN 0037        500 CONTINUE
  ISN 0038            STOP
  ISN 0039            END
```

FOUND FIELD WITH THE FOLLOWING IDENTS...
00100800 01117082 00000000 00000000 0000011D 00000A87 4E010200 0A0014F5 2A1AC45C 43829400 00000009 00000000

SELECTED 700 MB HEIGHTS (METERS) OVER THE UNITED STATES. (LOWER LEFT VALUE AT 30N, 125W)

3049. 3057. 3062. 3062. 3056. 3041. 3018. 2985. 2943. 2898. 2856. 2822. 2801. 2789. 2783. 2779. 2775. 2773. 2777.

3033. 3041. 3049. 3056. 3059. 3054. 3039. 3014. 2982. 2945. 2909. 2877. 2853. 2835. 2820. 2808. 2796. 2788. 2786.

3028. 3035. 3043. 3053. 3060. 3061. 3052. 3035. 3011. 2984. 2953. 2922. 2893. 2868. 2848. 2830. 2813. 2800. 2795.

3036. 3042. 3049. 3056. 3063. 3065. 3060. 3050. 3033. 3011. 2984. 2953. 2924. 2898. 2875. 2854. 2836. 2822. 2819.

3052. 3059. 3062. 3065. 3068. 3069. 3066. 3060. 3048. 3030. 3007. 2982. 2956. 2933. 2911. 2891. 2874. 2865. 2868.

3069. 3075. 3077. 3077. 3077. 3075. 3071. 3066. 3058. 3047. 3033. 3016. 2998. 2978. 2959. 2943. 2932. 2929. 2932.

3083. 3088. 3089. 3090. 3089. 3085. 3078. 3070. 3066. 3064. 3061. 3054. 3042. 3027. 3014. 3005. 3000. 2998. 2998.

3096. 3098. 3098. 3100. 3100. 3096. 3088. 3081. 3078. 3080. 3084. 3084. 3080. 3073. 3068. 3065. 3062. 3058. 3054.

3106. 3105. 3104. 3106. 3108. 3108. 3104. 3099. 3096. 3095. 3098. 3103. 3107. 3110. 3111. 3110. 3108. 3105. 3102.

VALID 0 HOURS AFTER 78/ 1/ 2, 000 GMT.

..... RAN TO PLANNED EXIT .....

Built-in Functions

IV = LAND (a, b)

where a,b may be a 1-, 2-, or 4-byte logical integer expression. The value of LAND is obtained by AND-ing the individual bits of the arguments. The resulting value, IV, will be considered to be logical*4, but may be used as an integer.

IV = LOR (a, b)

where a, b may be a 1-, 2-, or 4-byte logical or integer expression. The value of LOR is obtained by OR-ing the individual bits of the arguments. The resulting value, IV, will be considered to be logical*4, but may be used as an integer.

IV = LXOR (a, b)

where a, b may be a 1-, 2-, or 4-byte logical or integer expression. The value of LXOR is obtained by exclusive OR-ing the individual bits of the arguments. The resulting value, IV, will be considered to be logical*4, but may be used as an integer.

IV = SHFTL (J, K)

IV = SHFTR (J, K)

where J is a 4-byte variable,

K is the actual number of bits to be shifted.

The values of SHFTL and SHFTR are obtained by shifting the first argument, J, left or right, respectively, the number of bits specified by the second argument, K. The resulting value, IV, will be considered to be logical*4, but may be used as an integer.

IV = TBIT (A, K)

where A is a variable, 4-bytes or less,

K is the number assigned to the bit to be tested

The value of TBIT is .TRUE. or .FALSE. depending whether bit position K of the variable A is ON or OFF (ON=1, OFF=0). Bit 0 is the leftmost bit of variable A. The result, IV, will be declared as logical*4.

Intrinsic Functions

AMAX1 - Maximum, Real

$A=AMAX1(X_1,X_2,\ldots,X_n)$

where A and X are real *4

AMIN1 - Minimum, Real

$A=AMIN1(X_1,X_2,\ldots,X_n)$

where A and X are Real *4

MAX0 - Maximum, Integer

$L=MAX0(M_1, M_2,\ldots,M_n)$

where L and M are Integer *4

MIN0 - Minimum, Integer

$L=MIN0(M_1,M_2,\ldots,M_n)$

where L and M and Integer *4

IABS - Absolute Value, Integer

L = IABS(M)

where L and M are Integer *4

SIGN -

$Y=SIGN(X_1,X_2)$

where X and Y are Real *4

$Y=(\text{sign of } X_2).|X_1|$