

Model Change Bulletin

MCB#1

01/19/94

BPIP (dated 93320)

This Model Change Bulletin essentially initiates the activities of the SCRAM BBS for this model.

BPIP (Building Profile Input Program) is designed to calculate building heights and projected building widths based on an implementation of Good Engineering Practice Stack Height and building downwash guidance. The output data from BPIP has to be manually edited into the appropriate ISC2 input runstream.

This program will henceforth be identified by its name and a Julian date, i.e.: BPIP (dated 93320). Future changes to this program will be identified by an updated Julian date found on the first line of the model source code and followed by a reference to a SCRAM BBS Model Change Bulletin (MCB#n.txt) which will explain the nature and contents of the changes. All Model Change Bulletins for this model will always be available for back reference on this BBS. In the future, notification of changes (MCBs) to this model will be made in the SCRAM ALERTS section of this BBS immediately following the LOGON procedure.

The source code for this model has been compiled and tested on an IBM PC computer. A math coprocessor is required.

```
*****  
* Important Note *  
*****
```

New Model Change Bulletins will ONLY be issued when there is a change to the source code for this model. Changes to test cases and/or any pre and post processors that are included in the archived file for this model will be addressed and appropriately identified in an amendment to the model's latest MCB with notification posted in the SCRAM ALERTS section of the BBS.

Model Change Bulletin

MCB#2

BPIP (dated 94074)

This is a formal notification of changes made to the Building Profile Input Program (BPIP) source code. These changes result from found and reported problems with the program. The problems included:

1. No conversion of non-metric units to metric for the stack and building base elevations.
2. Slow processing speed for sufficiently close buildings.
3. An error where a Y coordinate was equated to a X coordinate when determining whether a stack is within 5L downwind of a Gap Filling Structure.
4. The use of the wrong variable with the right value in an ISCLT2 related algorithm.
5. The use of a subscript of a tier being combined with a focal tier instead of the focal tier subscript.

There were additional findings that lead to:

1. The editing of some of the explanations and identifications terms.
2. A revamping of the way the values for the direction-related parts of the Summary Table are calculated.
3. The changing of the Preliminary GEP stack height values from emission limitation values to design height values.

The following is a list of the effects that this change has on the meaning of and the values calculated:

1. The program will now properly account for the difference between stack and building base elevations when calculating the design GEP stack height.
2. Except for the 180 degree directions for test case 2 of MCB#1 and #2, the test case 1 and 2 building heights and projected widths will have the same respective output values. The difference in test case 2 is caused by how wake effect height values are stored in computer memory during a run. One wake effect height was previously stored as 18.2500004 while the other is currently stored as 18.2499995. Both values are suppose to be the same. This is an artifact of how floating point numbers are stored in computer memory. In cases where a set of wake effect heights are the same, but are produced by different building height and projected width values, the wake effect height produced using the narrower width should cause its associated building height and projected building width values to be saved. Because of the artifact, this was not the case in the previous version of BPIP (MCB#1). A slight change in how wake effect heights are compared, now produces the correct results.
3. In the Dominate Combined Building section of the Summary Table, if the maximum wake effect height for a particular direction was caused by a single tier, the single tier number was displayed. The single tier was displayed in both the Dominate Stand-alone Tier and Combined Building sections. This has been changed so that in the Dominate Combined Building section, no single tier numbers are displayed. If any tier numbers are displayed, they are the tier numbers of the combined buildings producing the highest wake effect height of all the combined building sets for that direction; even if the associated wake effect height is not the highest value.

The following is a list of the changes made to BPIP along with a brief explanation:

1. The BPIP SCRAM header was updated by changing the comment statements from:

```

C *
C *          BUILDING PROFILE INPUT PROGRAM (DATED 93320)
C *
C *          *** SEE BPIP MODEL CHANGE BULLETIN MCB#1 ***
C *

```

to:

```

C *
C *          BUILDING PROFILE INPUT PROGRAM (DATED 94074)
C *
C *          *** SEE BPIP MODEL CHANGE BULLETIN MCB#2 ***
C *

```

2. Several definitions were either added, deleted, or modified:

a. Those added:

```

C CNVFLG - FLAG TO INDICATE A CONVERSION FACTOR OF 1.00 OR NOT
C DE      - CURRENT DIRECTION - LAGS ACTUAL DIRECTION - USED IN MXBWH
C DFLG    - TO INDICATE A COMBINED STRUCTURE EXIST FOR A STACK - DIRECTION
C          COMBINATION
C TC      - NUMBER OF TIERS IN COMBINATION (MIN = 1)

```

b. Those modified:

```

C G65     - GEP 65 METER DECISION POINT FOR GEP DETERMINATION

```

c. Those deleted:

```

C MDIRS   - WIND FLOW DIRECTION MAXIMUM WAKE EFFECT HEIGHT OCCURS IN

```

3. The variables, CNVFLG and DE, were added and defined as integers while the variables AA and BB were added and defined as double precision. The resultant lines of code are shown:

```

INTEGER      IHR, IMIN, ISEC, IX, IYR, IMON, IDAY, C, C1, C2, CH,
*            CNVFLG, D, DE, GEPIN, GTLIST, GTNUM, MTNUM, MTLIST, S,
*            T1, T2, TL1, TL2, TLIST, TLIST2, TN1, TNUM, TNUM2, UTM

DOUBLE PRECISION AA, BB

```

4. The variables, DE and DFLG, were added to reset initialization values from one sector to the next and to identify those situations where a combined building is upwind of a stack for a particular direction. DE is initialized in the main body of the program and used in MXBWH while DFLG is set in MXBWH and then used in the main body of the program.

```

COMMON /DE/ DE, DFLG(MSK, MD)

```

5. The Labeled Common statements /HT/, /PBH/, and /STK/ are no longer needed. The variables TH, SH, XS, and YS were moved to other Dimension statements.

```

COMMON /HT/ TH(MB, MT), SH(MSK)
COMMON /PWH/ PBH(MBT), PBW(MBT), HWE(MBT)
COMMON /STK/ XS(MSK), YS(MSK)
COMMON /TNM/ TNUM2(MBT), TLIST2(MBT,MBT)

```

6. The variable MDIRS is no longer needed and was removed from the Labeled Common statement, /MTNM/.

```

COMMON /MTNM/ MTNUM(MSK), MTLIST(MSK,MBT), MDIRS(MSK)

```

7. The variables from comment 5 above were placed in the following Dimension statements:

```

DIMENSION BTN(MB), NTRS(MB), W(MBT), HT(MBT), TH(MB, MT)
DIMENSION STKN(MSK), SH(MSK), XS(MSK), YS(MSK)

```

8. The G65 statement was moved from well within the program to an area at the

beginning of the program where other variables are initialized. The new variable, DE, is also initialized in this same area. The code appears as follows:

```
DE = 0
DTR = 3.141593 / 180
G65 = 65.
```

9. Several lines of code were modified to improve the output spacing and appearance. The following were changed:

```
from:
WRITE(12,297)
WRITE(12,*) '===== '
to:
WRITE(12,*) '===== '

from:
WRITE(12,*) '===== '
WRITE(12,297)
WRITE(14,297)
WRITE(14,*) '===== '
to:
WRITE(12,*) '===== '
WRITE(14,*) '===== '

from:
WRITE(14,*) '===== '
WRITE(14,297)
to:
WRITE(14,*) '===== '
```

10. The following lines of code did not cover all variations of input. The following lines of code were changed from:

```
IF(SWTN .EQ. 'st') SWT = 0
IF(SWTN .EQ. 'ST') SWT = 0
IF(SWTN .EQ. 'lt') SWT = 1
IF(SWTN .EQ. 'LT') SWT = 1
WRITE(12,2) SWTN
WRITE(14,2) SWTN
to:
IF(SWTN(1:1) .EQ. 's') SWT = 0
IF(SWTN(1:1) .EQ. 'S') SWT = 0
IF(SWTN(1:1) .EQ. 'l') SWT = 1
IF(SWTN(1:1) .EQ. 'L') SWT = 1
IF(SWT .EQ. 0) THEN
WRITE(12,2) SWTN
WRITE(14,2) SWTN
ENDIF
IF(SWT .EQ. 1) THEN
WRITE(12,9) SWTN
WRITE(14,9) SWTN
ENDIF
```

11. The wording in Format statement #3 was improved which resulted in changes to the following two write statements. A flag was also added to identify when metric versus non metric input is being entered. The lines of code were changed from:

```
WRITE(12,3) UNTS, UNTS, CONV
WRITE(14,3) UNTS, UNTS, CONV
to:
WRITE(12,3) UNTS, CONV
WRITE(14,3) UNTS, CONV
C IF CNVFLG IS 1, THE UNITS ARE CONSIDERED TO BE IN METERS
CNVFLG = 0
IF (ABS(CONV - 1.00) .LT. .05) CNVFLG = 1
```

12. The following lines of code did not cover all variations of input. The following lines of code were changed from:

```
IF(UTMP .EQ. 'utmn') UTM = 0
IF(UTMP .EQ. 'UTMN') UTM = 0
IF(UTMP .EQ. 'utmy') UTM = 1
IF(UTMP .EQ. 'UTMY') UTM = 1
```

to:

```
IF(UTMP(4:4) .EQ. 'n') UTM = 0
IF(UTMP(4:4) .EQ. 'N') UTM = 0
IF(UTMP(4:4) .EQ. 'y') UTM = 1
IF(UTMP(4:4) .EQ. 'Y') UTM = 1
```

13. Format statement #11 was added as a result of editing Format statement #4. The result is the inclusion of another Write statement as shown below this first statement:

```
WRITE(14,4) UTMP
WRITE(14,11)
```

14. To improve output appearance the following statement was added in two places in the input area:

```
WRITE(12,297)
```

15. The conversion from non metric to metric base elevations has been added to this version. The coding was changed from:

```
WRITE(14,12) BTN(I), NTRS(I), BELEV(I), UNTS
```

to:

```
WRITE(14,12) BTN(I), NTRS(I), BELEV(I), UNTS
IF (CNVFLG .EQ. 0) THEN
  BELEV(I) = BELEV(I) * CONV
  WRITE(14,13) BELEV(I)
END IF
WRITE(14,14)
```

from:

```
WRITE(14,13) BTN(I), J, C, TH(I,J), ND(I,J)
```

to:

```
WRITE(14,21) BTN(I), J, C, TH(I,J), ND(I,J)
IF (CNVFLG .EQ. 0) THEN
  TH(I,J) = TH(I,J) * CONV
  WRITE(14,22) TH(I,J)
END IF
```

16. Double precision was added so the precision of the entered UTM coordinates could be maintained. The code was changed from:

```
READ(10,*) X(I, J, K), Y(I, J, K)
```

to:

```
READ(10,*) AA, BB
IF (CNVFLG .EQ. 1 .OR. UTM .EQ. 1) THEN
  WRITE (14,32) AA, BB
END IF
IF (CNVFLG .EQ. 0 .AND. UTM .EQ. 0) THEN
  WRITE (14,31) AA, BB, UNTS
  AA = AA * CONV
  BB = BB * CONV
  WRITE (14,32) AA, BB
END IF
```

from:

```
UEAST = X(1,1,1)
UNORTH = Y(1,1,1)
A = UEAST
```

to:

```

        UEAST = AA
        UNORTH = BB
        A = UEAST

from:

        ELSE
        A = X(I, J, K)
        B = Y(I, J, K)
        AU = A - UEAST

to:

        ELSE
        A = AA
        B = BB
        AU = A - UEAST

from:

        END IF
        WRITE (14,14) A, B
        WRITE (14,16) AU, BU
        X(I, J, K) = AU

to:

        END IF
        WRITE (14,33) AU, BU
        X(I, J, K) = AU

from:

        Y(I, J, K) = BU
        ELSE
        WRITE (14,14) X(I, J, K), Y(I, J, K)
        END IF

to:

        Y(I, J, K) = BU
C      WRITE (14,14) X(I, J, K), Y(I, J, K)
        ELSE
        X(I, J, K) = AA
        Y(I, J, K) = BB
        END IF

```

17. The variables, A and B were renamed to AP and BP to distinguish them from the variables A and B above. The code was changed from:

```

        IF (PN .EQ. 1) THEN
        A = X(I, J, K) * CSA + Y(I, J, K) * SNA
        B = Y(I, J, K) * CSA - X(I, J, K) * SNA
        X(I, J, K) = A
        Y(I, J, K) = B
        WRITE (14,19) X(I, J, K), Y(I, J, K)

to:

        IF (PN .EQ. 1) THEN
        AP = X(I, J, K) * CSA + Y(I, J, K) * SNA
        BP = Y(I, J, K) * CSA - X(I, J, K) * SNA
        X(I, J, K) = AP
        Y(I, J, K) = BP
        WRITE (14,19) X(I, J, K), Y(I, J, K)

```

18. The PRINT \* statement was deleted.

19. Conversion factor statements were added to the stack data processing section. The code was changed from:

```

        WRITE (14,11) NS

```

```

        WRITE (14,21)
DO 40 S = 1, NS
  READ (10,*) STKN(S), SB(S), SH(S), XS(S), YS(S)
    AU = XS(S) - UEAST
    BU = YS(S) - UNORTH

  IF (UTM .EQ. 0) THEN
    WRITE (14, 22) STKN (S), SB(S), SH(S), XS(S), YS(S)
  ELSE
    WRITE (14, 22) STKN(S), SB(S), SH(S), XS(S), YS(S)
    WRITE (14, 23) AU, BU
  END IF

```

to:

```

        WRITE (14,41) NS
        WRITE (14,42)
DO 40 S = 1, NS
  READ (10,*) STKN(S), SB(S), SH(S), AA, BB
    WRITE (14, 43) STKN(S), SB(S), SH(S), UNTS
      SB(S) = SB(S) * CONV
      SH(S) = SH(S) * CONV
    IF (CNVFLG .EQ. 0) WRITE(14,44) SB(S), SH(S)

    IF (CNVFLG .EQ. 1 .OR. UTM .EQ. 1) WRITE(14,51) AA, BB
    IF (CNVFLG .EQ. 0 .AND. UTM .EQ. 0) THEN
      WRITE (14,49) AA, BB, UNTS
      AA = AA * CONV
      BB = BB * CONV
      WRITE (14, 45) AA, BB
    END IF

    AU = AA - UEAST
    BU = BB - UNORTH

  IF (UTM .EQ. 1) THEN
    WRITE (14, 45) AU, BU
  END IF

```

20. A write statement was renumbered from:

```

        YS(S) = B
        WRITE (14,42) XS(S), YS(S)
    END IF

```

to:

```

        YS(S) = B
        WRITE (14,46) XS(S), YS(S)
    END IF

```

21. A comment line was edited. The line was changed from:

```

C          IDENTIFY TIER GROUPS EXAMINE FOR COMBINING

```

to:

```

C          IDENTIFY TIER GROUPS - EXAMINE FOR COMBINING

```

22. In several of the CALL GPC argument lists, the argument C2 should have been C1. Also, to improve the execution speed of the program, all the GO TO 136 lines were commented back into the code. The affected lines were changed from:

```

C          CALL GPC (D, I, C2, S, TW, WS, HTA, TL1)
          GO TO 136
    END IF

```

to:

```

          CALL GPC (D, I, C1, S, TW, WS, HTA, TL1)
          GO TO 136
    END IF

```

23. A second LFLAT IF statement was omitted from the original code. The code should read:

```
CALL DISLIN(X1, Y1, X2, Y2, L5, IBET, XPSTK, YPSTK)
IF (LFLAT (S,C2) .EQ. 1) IBET = 1
```

24. The line:

```
Y12 = XC(C1, K1)
```

was corrected to read:

```
Y12 = YC(C1, K1)
```

25. The G65 line was modified and moved to the initialization section near the beginning of the code:

```
G65 = 65.
```

26. The Preliminary GEP stack height was set for calculating an emissions limitation height. The new code is designed to calculate the GEP design height. The code was changed form:

```
WRITE(12, 297)
WRITE(12,1005) UNTS

WRITE(14, 1020) UNTS
DO 1010 S = 1, NS
IF ((SH(S) .GT. GEP(S)) .AND. (GEP(S) .EQ. 0.00)) PV = G65
IF (SH(S) .LE. G65) PV = SH(S)
IF ((SH(S) .GT. G65) .AND. (SH(S) .LT. GEP(S))) PV = SH(S)
IF ((SH(S) .GT. GEP(S)) .AND. (GEP(S) .GT. 0.00)) PV = GEP(S)
WRITE(14,1022) S, STKN(S), SH(S), PV, GEPBH(S), GEPBW(S),GEP(S)
IF(GTNUM(S) .EQ. 0) THEN
  WRITE(14,*) '      No tiers affect this stack.'
ELSE
```

to:

```
WRITE(12, 297)
WRITE(12,1005)

WRITE(14, 1020)
DO 1010 S = 1, NS
IF(GTNUM(S) .EQ. 0) THEN
  PV = G65
  WRITE(14,1022) S, STKN(S), SH(S), PV, GEPBH(S), GEPBW(S),
*      GEP(S)
  WRITE(14,*) '      No tiers affect this stack.'
  WRITE(12,1001) STKN(S), SH(S), GEP(S), PV
ELSE
```

from:

```
DIF = SB(S) - BELEV(MI(S))
WRITE(14,1025) DIF
WRITE(14,1023) GTNUM(S), GDIRS(S)
WRITE(14,1024) (GTLIST(S,I), I = 1, GTNUM(S))
END IF
```

to:

```
DIF = SB(S) - BELEV(MI(S))
GP = GEP(S) - DIF
PV = MAX (G65, GP)
* WRITE(14,1022) S, STKN(S), SH(S), PV, GEPBH(S), GEPBW(S),
  GEP(S)
WRITE(14,1025) DIF
WRITE(14,1023) GTNUM(S), GDIRS(S)
WRITE(14,1024) (GTLIST(S,I), I = 1, GTNUM(S))
WRITE(12,1000) STKN(S), SH(S), DIF, GEP(S), PV
```

END IF

27. As a result of comment 26, the following statement was deleted:

```
WRITE(12,1000) STKN(S), SH(S), GEP(S), PV
```

28. The BPIP output is in meters. The following statement was changed from:

```
WRITE(14, 2020) UNITS
```

to:

```
WRITE(14, 2020)
```

29. To conserve variable names in a labeled common statement, a statement was added prior to the call to MXBWH. The code was changed from:

```
C          If so, calculate the PBW & PBH, save max values
CALL MXBWH(D, I, S, C, TW, HTA, WS, CH)
```

to:

```
C          If so, calculate the PBW & PBH, save max values
          TLIST2(C,1) = C
CALL MXBWH(D, I, S, C, TW, HTA, WS, CH)
```

30. The variable, MI(S), which does not have a directional subscript, was replaced by the variable, MHWE(S,D), which does have a directional subscript. Also, the wording in the write statement was edited. The code was changed from:

```
IF (MI(S) .EQ. 0) THEN
  WRITE(14,*) '      No tier affects this stack.'
ELSE
```

to:

```
IF (MHWE(S,D) .EQ. 0.0) THEN
  WRITE(14,*) '      No single tier affects this stack for this
&direction.'
ELSE
```

31. When only one building is being processed, the 'Dominate combined building' section is not needed. The coding was changed from:

```
WRITE(14,*) 'Dominate combined buildings:'
DO 350 D = 1, NDIR
```

to:

```
IF (NB .EQ. 1) THEN
  WRITE(14,*) 'Dominate combined buildings: None'
ELSE
  WRITE(14,*) 'Dominate combined buildings:'
END IF
DO 350 D = 1, NDIR
```

from:

```
ANG = D * DDEG
WRITE(14,604) ANG
ANG = ANG * DTR
```

to:

```
ANG = D * DDEG
IF (NB .GT. 1) WRITE(14,604) ANG
ANG = ANG * DTR
```

from:

```

430 CONTINUE
DO 2012 S = 1, NS

```

to:

```

430 CONTINUE
IF(NB .GT. 1) THEN
DO 2012 S = 1, NS

```

32. The directional data in the summary table was not always clear as to its meaning. Sometimes a single dominate tier was listed in the dominate combined section of the summary table. The following keeps the dominate stand-alone and combined building section 'directional MAX' data separate. The code was changed from:

```

IF (MTNUM(S) .EQ. 0) THEN
WRITE(14,*) ' No tiers affect this stack.'
ELSE
DIF = SB(S) - BELEV(MI(S))
WRITE(14,1025) DIF
WRITE(14,2023) MTNUM(S)
WRITE(14,2024) (MTLIST(S,M), M = 1, MTNUM(S))
END IF

```

to:

```

IF (MTNUM(S) .EQ. 0) THEN
WRITE(14,*) ' No combined tiers affect this stack for th
&is direction.'
ELSE
IF (DFLG(S,D) .EQ. 1) THEN
DIF = SB(S) - BELEV(MI(S))
WRITE(14,1025) DIF
WRITE(14,2023) MTNUM(S)
WRITE(14,2024) (MTLIST(S,M), M = 1, MTNUM(S))
ELSE
WRITE(14,*) ' No combined tiers affect this stack for
& this direction'
END IF
END IF

```

and from:

```

2012 CONTINUE
350 CONTINUE

```

to:

```

2012 CONTINUE
END IF
350 CONTINUE

```

33. The BPIP output needs to be in meters. The second write statement was changed from:

```

WRITE (12, 1) TITLE
WRITE (12, *) ' BPIP output in ', UNITS

```

to:

```

WRITE (12, 1) TITLE
WRITE (12, *) ' BPIP output is in meters'

```

34. In the second line, the MBT variable contains the right value but is the wrong variable. MBT was replaced by NDIR. The code was changed from:

```

IF (L .GT. ML) L = L - ML
IF (J .GT. MBT) J = J - MBT
IF (MHWE(S, D) .GT. 0.0) THEN

```

to:

```
IF (L .GT. ML) L = L - ML
IF (J .GT. NDIR) J = J - NDIR
IF (MHWE(S, D) .GT. 0.0) THEN
```

35. A number of the format statements were modified to improve the meaning and appearance. Two statements were also corrected. The code was changed from:

```
*ST2 run.*/)
3  FORMAT(3X,'The inputs units are in: ', A10,' Multiplying ',A10,' b
*y a conversion '/3X,' factor of',F10.4, ' will produce internal BP
*IP results in meters.*/)
4  FORMAT(3X,'The UTMP variable is set to ',A4,'. The input is assum
```

to:

```
*ST2 run.*/)
3  FORMAT(3X,'Inputs entered in ', A10,' will be converted to meters
*using '/3X,' a conversion factor of',F10.4, '. Output will be in
*meters.*/)
4  FORMAT(3X,'The UTMP variable is set to ',A4,'. The input is assum
```

from:

```
* UTM coordinates entered to form '/4X,' this new local coordinate
*system.'//3X,'The new local coordinates will be displayed in paren
*theses just below'/4X,' the UTM coordinates they represent.',//)
5  FORMAT(3X,'UTMP is set to ',A4,'. The input is assumed to be in a
```

to:

```
* UTM coordinates entered to form '/4X,' this new local coordinate
*system.'/)
5  FORMAT(3X,'UTMP is set to ',A4,'. The input is assumed to be in a
```

from:

```
* local'/3x,' X-Y coordinate system as opposed to a UTM coordinate
*system.'/3x,' True North is in the positive Y direction.'//)
6  FORMAT(1X,'Number of buildings to be processed :',I4/)
7  FORMAT(37X,'(',2F12.2,')')
```

to:

```
* local'/3x,' X-Y coordinate system as opposed to a UTM coordinate
*system.'/3x,' True North is in the positive Y direction.'/)
6  FORMAT(1X,'Number of buildings to be processed :',I4)
7  FORMAT(37X,'(',2F12.2,')')
```

from:

```
8  FORMAT(' Factor to convert from input units to meters is:',F10.4)
11 FORMAT(/1X,'Number of stacks to be processed :',I4/)
12 FORMAT(1X,A8,' has',I2,' tier(s) with a base elevation of', F8.2,
*' ',A10, //
*' BUILDING TIER BLDG-TIER TIER NO. OF CORNER COORDINA
*TES',
*/' NAME NUMBER NUMBER HEIGHT CORNERS X Y')
```

to:

```
8  FORMAT(' Factor to convert from input units to meters is:',F10.4)
9  FORMAT(/3X,'The ',A2,' flag has been set for processing for an ISC
*LT2 run.*/)
11 FORMAT(3X,'The new local coordinates will be displayed in parenthe
*ses just below'/4X,' the UTM coordinates they represent.',/)
12 FORMAT(/1X,A8,' has',I2,' tier(s) with a base elevation of', F8.2
*,' ',A10)
13 FORMAT(47X,'(',F8.2,') meters',/)
```

```

14  FORMAT(
    *' BUILDING TIER BLDG-TIER TIER NO. OF CORNER COORDINAT
    *ES',
    */' NAME NUMBER NUMBER HEIGHT CORNERS X Y'

from:

13  FORMAT(1X,A8,I5,5X,I4,4X,F6.2,I6)
14  FORMAT(42X,2F12.2)
16  FORMAT(41X,'(',2F12.2,')')
17  FORMAT(3X,'Plant north is',F7.2,' degrees with respect to True Nor
    *th. ')
18  FORMAT(4X,'The plant coordinates will appear as entered in the Sum

to:

    */)
17  FORMAT(3X,'Plant north is set to',F7.2,' degrees with respect to T
    *rue North. '//)
18  FORMAT(4X,'The plant coordinates will appear as entered in the Sum

from:

    *elow between'/4X,'the square brackets.')
19  FORMAT(41X,['(',2F12.2,']')
21  FORMAT('
    *   STACK
    *   STACK NAME BASE HEIGHT X COORDINATES'/
    *   Y'//)
22  FORMAT (2X, A8,3X, 2F8.2, 2X, 2F12.2)
23  FORMAT (30X,'(',2F12.2,')')
42  FORMAT(30X,['(',2F12.2,']')
47  FORMAT(' Caution: Blank spaces are not allowed in Stack names by

to:

    *elow between'/4X,'the square brackets.')
19  FORMAT(41X,['(',2F12.2,'] meters')
21  FORMAT(1X,A8,I5,5X,I4,4X,F6.2,I6)
22  FORMAT(27X,F6.2,' meters')
31  FORMAT(42X,2F12.2, 1X, A10)
32  FORMAT(42X,2F12.2, ' meters')
33  FORMAT(41X,'(',2F12.2,') meters')
41  FORMAT(/1X,'Number of stacks to be processed :',I4/)
42  FORMAT('
    *   STACK
    *   STACK NAME BASE HEIGHT X COORDINATES'/
    *   Y'//)
43  FORMAT (2X, A8,3X, 2F8.2, 1X, A10)
44  FORMAT (12X,'(', 2F8.2, ') meters')
45  FORMAT (30X,'(', 2F12.2,') meters')
46  FORMAT (30X,['(', 2F12.2,'] meters')
47  FORMAT(' Caution: Blank spaces are not allowed in Stack names by

from:

    *ISC2 models.')
71  FORMAT (//' The number of stack-tier combinations entered, where e

to:

    *ISC2 models.')
49  FORMAT(31X,2F12.2, 1X, A10)
51  FORMAT (31X, 2F12.2,' meters')
71  FORMAT (//' The number of stack-tier combinations entered, where e

from:

604  FORMAT(/1X,'Drctn: ', F6.2/)
1000  FORMAT(15X, A8,2X, 3(F8.2,2X))
1005  FORMAT(15X,'PRELIMINARY* GEP STACK HEIGHT RESULTS TABLE'/
    * 12X,' (Input Units: ',A10,')'//
    *15X,' Preliminary*'/
    *15X,' Stack Stack GEP** GEP Stack'/
    *15X,' Name Height EQN1 Height Value'//)

```

```

1007 FORMAT(/' * Results based on Table 3.1 of the GEP Technical Supp
*ort Document.'/
*' Consult Table 3.1 for any additional steps that may be requi
*red.'/
*' ** Results using Equation 1, page 6 of GEP Technical Support Do
*cument.')

```

```

1020 FORMAT(//' Overall GEP Summary Table'/
* '(Units: ', A8,')'//)

```

```

1021 FORMAT(10X,'NOTE: The projected width values below are not always'
to:

```

```

604 FORMAT(/1X,'Drctn: ', F6.2/)
1000 FORMAT(8X, A8, 4(F8.2,5X))
1001 FORMAT(8X, A8, F8.2, 10X, 'N/A',5X, 3(F8.2,5X))
1005 FORMAT(16X,'PRELIMINARY* GEP STACK HEIGHT RESULTS TABLE'/
* 13X,' (Output Units: meters)'//
*8X,' Stack-Building Preliminary*'/
*8X,' Stack Stack Base Elevation GEP** GEP Stack'/
*8X,' Name Height Differences EQN1 Height Value'//)
1007 FORMAT(/' * Results are based on Determinants 1 & 2 on pages 1 &
* 2 of the GEP' ' Technical Support Document. Determinant 3 m
*ay be investigated for'/' additional stack height credit. Fin
*al values result after'/' Determinant 3 has been taken into co
*nsideration.'/
*' ** Results were derived from Equation 1 on page 6 of GEP Techni
*cal'/' Support Document. Values have been adjusted for any st
*ack-building'/' base elevation differences.'//
*' Note: Criteria for determining stack heights for modeling e
*mission'/
*' limitations for a source can be found in Table 3.1 of the'/'
*' GEP Technical Support Document.'//)

```

```

1020 FORMAT(//' Overall GEP Summary Table'/
* '(Units: meters)'//)

```

```

1021 FORMAT(10X,'NOTE: The projected width values below are not always'
from:

```

```

1024 FORMAT(' Bldg-Tier nos. contributing to GEP:', 10I4)
1025 FORMAT(11X,'*with a Stack-Building elevation difference applied ='
*,F8.2)
2020 FORMAT(//' Summary By Direction Table'/
* '(Units: ', A8,')',
*// ' Dominate stand alone tiers:')

```

to:

```

1024 FORMAT(' Bldg-Tier nos. contributing to GEP:', 10I4)
1025 FORMAT(10X,'*adjusted for a Stack-Building elevation difference of
*',F8.2)
2020 FORMAT(//' Summary By Direction Table'/
* '(Units: meters)',
*// ' Dominate stand alone tiers:')

```

from:

```

2022 FORMAT(' StkNo:', I3, ' Stk Name:', A8, 23X, ' Stack Ht:', F8.2)
2026 FORMAT(11X, ' MAX: BH:',F7.2, ' PBW:',F7.2,
* ' *Wake Effect Ht:', F8.2)

```

to:

```

2022 FORMAT(' StkNo:', I3, ' Stk Name:', A8, 23X, ' Stack Ht:', F8.2)
2026 FORMAT(5X, 'Directional MAX: BH:',F7.2, ' PBW:',F7.2,
* ' *Wake Effect Ht:', F8.2)

```

36. The Labeled Common statement, /HT/, is no longer needed and was deleted. The code was changed from:

```

COMMON /ELEV/ BELEV(MB), SB(MSK)
COMMON /HT/ TH(MB, MT), SH(MSK)
COMMON /GP/ GEP(MSK), GEPBH(MSK), GEPBW(MSK), GEPIN(MSK,MBT,MBT)

```

to:

```
COMMON /ELEV/ BELEV(MB), SB(MSK)
COMMON /GP/ GEP(MSK), GEPBH(MSK), GEPBW(MSK), GEPIN(MSK,MBT,MBT)
```

37. The stack-building base elevation difference was brought forward in the GPC and MXBWH subroutines. The following sections of code were changed from:

```
HWE = HTA + 1.5 * WS
GEPIN(S, C, CH) = 1
```

to:

```
HWE = HTA + BELEV(I) - SB(S) + 1.5 * WS
GEPIN(S, C, CH) = 1
```

from:

```
IF (HWE .GT. GEP(S)) THEN
  GEP(S) = HWE + BELEV(I) - SB(S)
  GEPBH(S) = HTA
```

to:

```
IF (HWE .GT. GEP(S)) THEN
  GEP(S) = HWE
  GEPBH(S) = HTA
```

from:

```
IF (TW .LT. GEPBW(S)) THEN
  GEP(S) = HWE + BELEV(I) - SB(S)
  GEPBH(S) = HTA
```

to:

```
IF (TW .LT. GEPBW(S)) THEN
  GEP(S) = HWE
  GEPBH(S) = HTA
```

38. Two variables, DE and DFLG, were added through the Labeled Common statement, /DE/, to MXBWH. See comment #4 above for details. The code in MXBWH was changed from:

```
COMMON /ELEV/ BELEV(MB), SB(MSK)
```

to:

```
COMMON /DE/ DE, DFLG(MSK, MD)
COMMON /ELEV/ BELEV(MB), SB(MSK)
```

39. The Labeled Common statements, /HT/ and /PWH/, are no longer needed and resulted in the following statements being changed from:

```
COMMON /GP/ GEP(MSK), GEPBH(MSK), GEPBW(MSK), GEPIN(MSK,MBT,MBT)
COMMON /HT/ TH(MB, MT), SH(MSK)
COMMON /MXB/ MHWE(MSK, MD), MXPBH(MSK, MD), MXPBW(MSK, MD)
COMMON /PWH/ PBH(MBT), PBW(MBT), HWE(MBT)
COMMON /MIJ/ MI(MSK), MJ(MSK)
```

to:

```
COMMON /GP/ GEP(MSK), GEPBH(MSK), GEPBW(MSK), GEPIN(MSK,MBT,MBT)
COMMON /MXB/ MHWE(MSK, MD), MXPBH(MSK, MD), MXPBW(MSK, MD)
COMMON /MIJ/ MI(MSK), MJ(MSK)
```

40. The variable, MDIRS, is no longer being used. The labeled Common statement was changed from:

```
COMMON /MTNM/ MTNUM(MSK), MTLIST(MSK,MBT), MDIRS(MSK)
```

to:

```
COMMON /MTNM/ MTNUM(MSK), MTLIST(MSK,MBT)
```

41. In the MXBHW subroutine, several changes were made to simplify the code and add direction-based results for the summary table output. For these results, several variable initialization lines of code were added so that the variables would be initialized as the sector changed.

The statements beginning with MI(S) and ending with the 578 continue statement are repeated in three sections. Also the stack-building base elevation difference calculation is repeated three times. Both can be calculated in one section or place. The code was changed from:

```
C      Stack is within GEP 5L ?
      IF (GEPIN(S,C,TL1) .EQ. 1) THEN
        PBH(C) = HTA
        PBW(C) = TW
        HWE(C) = HTA + 1.5 * WS
        IF (HWE(C) .GT. MHWE(S, D)) THEN
          MHWE(S, D) = HWE(C) + BELEV(I) - SB(S)
          MXPBH(S, D) = PBH(C)
          MXPBW(S, D) = PBW(C)
          MTNUM(S) = TNUM2(C)
          MI(S) = I
          MJ(S) = C - (I-1)*MT
          MDIRS(S) = FLOAT(D)
          DO 578 M = 1, MTNUM(S)
            MTLIST(S,M) = TLIST2(C,M)
578      CONTINUE
        END IF
C      When wake effects are equal, use those values with a lesser projected width.
      IF (HWE(C) .EQ. MHWE(S,D)) THEN
        IF (PBW(C) .LT. MXPBW(S,D)) THEN
          MHWE(S, D) = HWE(C) + BELEV(I) - SB(S)
          MXPBH(S, D) = PBH(C)
          MXPBW(S, D) = PBW(C)
          MTNUM(S) = TNUM2(C)
          MI(S) = I
          MJ(S) = C - (I-1)*MT
          MDIRS(S) = FLOAT(D)
          DO 579 M = 1, MTNUM(S)
            MTLIST(S,M) = TLIST2(C,M)
579      CONTINUE
        END IF
      END IF
C      When a wake effect height is greater than the GEP STK Ht, use the GEP values
      IF (GEP(S) .LT. MHWE(S,D)) THEN
        MHWE(S, D) = GEP(S)
        MXPBW(S,D) = GEPBW(S)
        MXPBH(S,D) = GEPBH(S)
        MTNUM(S) = TNUM2(C)
        MI(S) = I
        MJ(S) = C - (I-1)*MT
        MDIRS(S) = FLOAT(D)
        DO 580 M = 1, MTNUM(S)
          MTLIST(S,M) = TLIST2(C,M)
580      CONTINUE
      END IF
    END IF
C
C INITIALIZE VALUES FOR NEXT LOOP
C
      HWE(C) = 0.0
      PBH(C) = 0.0
      PBW(C) = 0.0
```

to:

```
C
C Stack is within GEP 5L ?
C
  IF (GEPIN(S,C,TL1) .EQ. 1) THEN
    PBH = HTA
    PBW = TW
    HWE = HTA + BELEV(I) - SB(S) + 1.5 * WS
  IF (DE .NE. D) THEN
    DE = D
    DHWE = 0.0
    DPBH = 0.0
    DPBW = 0.0
    DO 577 SS = 1, MSK
577   DFLG(SS, D) = 0
  END IF
  IF (HWE - MHWE(S, D) .GT. .001) THEN
    MHWE(S, D) = HWE
    MXPBH(S, D) = PBH
    MXPBW(S, D) = PBW
    MFLG = 1
  END IF
C When wake effects are equal, use those values with a lesser projected width.
  IF (ABS(HWE - MHWE(S,D)) .LT. .001) THEN
    IF (PBW .LT. MXPBW(S,D)) THEN
      MHWE(S, D) = HWE
      MXPBH(S, D) = PBH
      MXPBW(S, D) = PBW
      MFLG = 1
    END IF
  END IF
C When a wake effect height is greater than the GEP STK Ht, use the GEP values
  IF (GEP(S) .LT. MHWE(S,D)) THEN
    MHWE(S, D) = GEP(S)
    MXPBW(S,D) = GEPBW(S)
    MXPBH(S,D) = GEPBH(S)
    MFLG = 1
  END IF
C Update and retain data for the Summary by Direction Tables
  IF (MFLG .EQ. 1) THEN
    DFLG(S,D) = 1
    MTNUM(S) = TNUM2(C)
    MI(S) = I
    MJ(S) = C - (I-1)*MT
    DO 578 M = 1, MTNUM(S)
578   MTLIST(S,M) = TLIST2(C,M)
  CONTINUE
  MFLG = 0
  END IF
C Determine if any combined buildings exist for a particular sector
C Save the combination producing the highest direction-based HWE for
C the Summary Table
  IF (TNUM2(C) .GT. 1) THEN
    IF (HWE .GT. DHWE) THEN
      MFLG = 2
    END IF
    IF ((ABS(HWE-DHWE).LT. .001) .AND. (PBW .LT. DPBW)) THEN
      MFLG = 2
    END IF
    IF (MFLG .EQ. 2) THEN
      DFLG(S, D) = 1
      DHWE = HWE
      DPBW = PBW
      MFLG = 0
      MTNUM(S) = TNUM2(C)
      MI(S) = I
      MJ(S) = C - (I-1) * MT
      DO 579 M = 1, MTNUM(S)
```

```
579          MTLIST(S,M) = TLIST2(C,M)
          CONTINUE
        END IF
      ENDIF
    END IF
```

Model Change Bulletin

MCB#3

BPIP (dated 95039)

This is a formal notification of changes made to the Building Profile Input Program (BPIP) source code. These changes result from found and reported problems with the program and from the need to make several revisions. The changes are discussed first and are then followed by the changes made to the source code.

Revisions were also made to the BPIP User's Guide. These revisions only reflect the changes pertinent to the roof stack algorithm and the roof stack input section. No changes were made to the user's guide test cases nor were there any subsequent changes to the Output and Summary file sections.

DISCUSSION OF SOURCE CODE CHANGES

As part of the previous version of BPIP, the user was required to manually determine which stacks were atop which tiers and then enter roof stack values into the input file after the stack input section. This was in response to cases where a stack could be atop a tier and more than 5L from an upwind tier roof edge. An algorithm has been added to automatically detect when a stack is on top of a tier. The user is no longer required to enter roof stack data. However, the user must be certain that the corner coordinates for each tier are entered sequentially; following the tier perimeter in a clockwise or counter-clockwise direction.

The new roof stack algorithm determines the direction from a stack to each corner of a tier. The directions are used to calculate each angle formed by lines drawn from a stack to each of two consecutive corners of a tier. If an angle is formed in a clockwise direction from the first to second corner, the angle is considered positive. If the angle is formed in a counter-clockwise direction, the angle is considered negative. If the stack is on top of the tier, the sum of the angles for the tier equals the absolute value of 360 degrees. If the stack is not on top of the tier, the sum is equal to 0. The flag, LFLAT, is set to one if the stack is on the roof of a tier.

In producing the Summary file, several arrays and one variable were not initialized and reinitialized. This has led to situations where the same group of tiers, influencing a stack for one wind flow direction, are repeated for subsequent directions where there is no wake effect influence being exerted by that particular group. The variable and arrays are now being initialized and reinitialized.

In producing the preliminary GEP stack height values for GEP wake effects over 65 meters, the stack-terrain difference value was added twice; once in calculating the GEP value and again just before printing the GEP value. The problem has been corrected.

SOURCE CODE CHANGES

The following is a list of changes made to BPIP:

1. The SCRAM header has been changed to reflect the new BPIP version date. The header line has been changed from:

C \* BUILDING PROFILE INPUT PROGRAM (DATED 94074) \*

to:

C \* BUILDING PROFILE INPUT PROGRAM (DATED 95039) \*

2. The following comment has been added:

C February 8, 1995 - Corrections/Updates (See MCB #3)

3. The following input format structure for the old roof stack input has been deleted:

C Number of Stacks whose roof location > 5L from a roof edge  
C Stack number, Building number, Tier Number  
C .  
C .  
C .  
C Stack number, Building number, Tier Number  
C

- 4a. The following definitions have been added:

C AA - DOUBLE PRECISION INPUT VALUE; GENERALLY AN X-COORDINATE  
C - ANGLE FORMED BETWEEN TWO ADJACENT CORNERS OF A TIER AND A STACK  
C AP - X-COORDINATE TRANSLATED FROM PLANT NORTH  
C BB - DOUBLE PRECISION INPUT VALUE; GENERALLY AN Y-COORDINATE  
C BLDGIN - LABELLED COMMON CONTAINING ARRAYS OF BUILDING INPUT VALUES  
C BLDOUT - LABELLED COMMON CONTAINING TRANSLATED CORNER COORDINATES  
C BP - Y-COORDINATE TRANSLATED FROM PLANT NORTH  
C DHWE - COMBINED BUILDING WAKE EFFECT HEIGHT FOR A WIND FLOW DIRECTION  
C DIRT - DIRECTION FROM A STACK TO A CORNER  
C DIRTT - TOTAL OF ANGLES (AA) FORMED BETWEEN A STACK AND CONSECUTIVE TIER  
C CORNERS. IF TOTAL EQUALS 360 DEGREES, THE STACK IS ON THE ROOF  
C OF THAT TIER.  
C DPBH - COMMON HEIGHT OF A COMBINED TIERS FOR A SPECIFIC WIND FLOW  
C DPBW - PROJECTED WIDTH OF COMBINED TIERS FOR A SPECIFIC WIND FLOW  
C DTR2 - DOUBLE PRECISION DTR  
C DX1 - MINIMUM OF TWO X-COORDINATES  
C DX2 - MAXIMUM OF TWO X-COORDINATES  
C ELEV - LABELLED COMMON CONTAINING BUILDING AND STACK BASE ELEVATIONS  
C GETDAT - GET DATE MICROSOFT LIBRARY FUNCTION  
C GETTIM - GET TIME MICROSOFT LIBRARY FUNCTION  
C GP - LABELLED COMMON CONTAINING GEP DATA  
C GTNM - LABELLED COMMON CONTAINING TIERS AFFECTING STACKS  
C IG - FLAG TO INDICATE SINGLE OR COMBINED TIER CALCULATIONS  
C II - COUNTER  
C INTRCP - LABELLED COMMON CONTAINING THE INTERCEPT COORDINATES BETWEEN A  
C STACK AND THE SIDE OF A TIER  
C ISS - COUNTER FOR (RE)-SETTING STACK RELATED VALUES  
C L - COUNTER  
C LB - BUILDING NUMBER INDEX  
C LD - TIER CORNER INDEX  
C LD1 - FIRST OF TWO CONSECUTIVE TIER CORNERS  
C LD2 - SECOND OF TWO CONSECUTIVE TIER CORNERS  
C LS - STACK INDEX  
C LT - TIER INDEX  
C MFLG - FLAG INDICATES A NEW MAXIMUM WAKE EFFECT HEIGHT HAS BEEN FOUND  
C MIJ - LABELLED COMMON CONTAINING BUILDING TIER NUMBER  
C MTNM - LABELLED COMMON CONTAINING TIER(S) PRODUCING MAXIMUM WAKE EFFECT  
C HEIGHT  
C MXB - LABELLED COMMON CONTAINING MAXIMUM WAKE EFFECT VALUES  
C MXN - LABELLED COMMON CONTAINING TIER MAX/MIN CORNERS  
C SWTN - OUTPUT FOR 'ST' OR 'LT' VARIABLE

```

C TL1 - FOCAL TIER NUMBER
C TL2 - CANDIDATE TIER NUMBER
C TNM - LABELLED COMMON CONTAINING A SET OF COMBINED TIERS
C UNTS - INPUT UNITS NAME
C UTM - FLAG TO INDICATE THE INPUT DATA ARE IN UTM COORDINATES
C UTMP - INPUT FLAG INDICATING USE OF UTM INPUT
C X1 - FIRST OF TWO CONSECUTIVE TIER CORNER X-COORDINATES
C X2 - SECOND OF TWO CONSECUTIVE TIER CORNER X-COORDINATES
C XCOMP - DOUBLE PRECISION DIFFERENCE BETWEEN A STACK AND TIER CORNER
C XSP - STACK X-COORDINATE ARGUMENT IN SUBROUTINE DISLIN
C Y1 - FIRST OF TWO CONSECUTIVE TIER CORNER Y-COORDINATES
C Y2 - SECOND OF TWO CONSECUTIVE TIER CORNER Y-COORDINATES
C YCOMP - DOUBLE PRECISION DIFFERENCE BETWEEN A STACK AND TIER CORNER
C YSP - STACK Y-COORDINATE ARGUMENT IN SUBROUTINE DISLIN

```

4b. The following variables and definitions are no longer being used and have been deleted.

```

C JXMAX - TIER CORNER NUMBER WHICH IS FURTHEST EAST
C JXMIN - TIER CORNER NUMBER WHICH IS FURTHEST WEST
C JYMAX - TIER CORNER NUMBER WHICH IS FURTHEST NORTH
C JYMIN - TIER CORNER NUMBER WHICH IS FURTHEST SOUTH
C TC - NUMBER OF TIERS IN COMBINATION (MIN = 1)

```

5. Several new and old variables were made double precision. These variables are involved with UTM coordinates and the new automated roof stack algorithm. The double precision statement was changed from:

```
DOUBLE PRECISION AA, BB
```

to:

```
DOUBLE PRECISION A, B, AA, BB, DIRT, DIRTT, DTR2,
+ UEAST, UNORTH, XCOMP, YCOMP
```

6. In several of the labelled common statement, variables were added or an additional subscript was added as part of the correction to keep a group of tiers from being printed for subsequent directions in the Summary table even though the group was no longer affecting the particular stack.

Previously, a maximum wake effect height printed for a particular direction in the Dominate Combined Building section could have been produced by a single tier. The variables DHWE, DPBH, and DPBW, have been added to keep a maximum wake effect height produced by a group of tiers separate from one produced by a single tier. Also, an extra argument has been added to the variable, GEPIN, in order to flag when a stack is in a GEP wake effect area produced by a group of tiers as opposed to a single tier. The labelled common statements are not contiguous.

The labelled common statements were changed from:

```
COMMON /DE/ DE, DFLG(MSK, MD)
COMMON /GP/ GEP(MSK), GEPBH(MSK), GEPBW(MSK), GEPIN(MSK,MBT,MBT)
```

```
COMMON /MIJ/ MI(MSK), MJ(MSK)
COMMON /MTNM/ MTNUM(MSK), MTLIST(MSK,MBT)
```

to:

```
COMMON /DE/ DE, DFLG(MSK, MD), DHWE(MSK), DPBH(MSK), DPBW(MSK)
COMMON /GP/ GEP(MSK), GEPBH(MSK), GEPBW(MSK),
+ GEPIN(MSK,MBT,MBT,2)
```

```
COMMON /MIJ/ MI(MSK,2), MJ(MSK,2)
COMMON /MTNM/ MTNUM(MSK,2), MTLIST(MSK,MBT,2)
```

7. Two variables were dimensioned for use with the roof stack algorithm. The following dimension statement was changed from:

```
DIMENSION DISTMN(MBT, MBT), LFLAT(MSK, MBT)
```

to:

```
DIMENSION DIRT(MTS), DISTMN(MBT, MBT), LFLAT(MSK, MBT)
```

8. A double precision degrees to radians conversion factor was added:

```
DTR2 = 3.141593 / 180.
```

9. A BPIP version header print statement was added using Format number 461. Original Format statement numbers 461 and 462 were increased by one. This was done in several places throughout the source code. The lines of code were changed from:

```
WRITE (12, 461) IMON, IDAY, IYR
WRITE (12, 462) IHR, IMIN, ISEC
WRITE (12,1) TITLE
WRITE (14, 461) IMON, IDAY, IYR
WRITE (14, 462) IHR, IMIN, ISEC
WRITE (14,1) TITLE
```

to:

```
WRITE (12, 461)
WRITE (12, 462) IMON, IDAY, IYR
WRITE (12, 463) IHR, IMIN, ISEC
WRITE (12,1) TITLE
WRITE (14, 461)
WRITE (14, 462) IMON, IDAY, IYR
WRITE (14, 463) IHR, IMIN, ISEC
WRITE (14,1) TITLE
```

and from:

```
560 CONTINUE
      WRITE (12, 461) IMON, IDAY, IYR
      WRITE (12, 462) IHR, IMIN, ISEC

461 FORMAT(1X,'DATE : ',I2,'/',I2,'/',I2)
462 FORMAT(1X,'TIME : ',I2,':',I2,':',I2)
```

to:

```
560 CONTINUE
      WRITE (12, 461)
      WRITE (12, 462) IMON, IDAY, IYR
      WRITE (12, 463) IHR, IMIN, ISEC

461 FORMAT(30X,'BPIP (Dated: 95039)')
462 FORMAT(1X,'DATE : ',I2,'/',I2,'/',I2)
463 FORMAT(1X,'TIME : ',I2,':',I2,':',I2)
```

10. VAX Fortran compilation warnings were given because of the way strings were continued from line to line. The strings in the following write and format statements were rewritten to VAX compilation standards. The following lines were changed from:

```
WRITE(*,*) 'The SWTN variable, ',SWTN,' is incorrectly enter
*ed.'
WRITE(*,*) 'The UTMP variable, ',UTMP,' is incorrectly enter
*ed.'
```

```
WRITE(*, *) 'THE NUMBER OF BUILDINGS ENTERED EXCEEDS THE PA
*RAMETER, MB'
WRITE(*, *) 'THE NUMBER OF TIERS/BUILDING EXCEEDS THE PARA
*METER, MT'
WRITE(*, *) 'THE NUMBER OF TIERS/BUILDING EXCEEDS THE PARA
*METER, MTS'
WRITE(*, *) 'THE NUMBER OF STACKS ENTERED EXCEEDS THE PARAMET
*ER, MKS'
```

```
2 FORMAT(/3X,'The ',A2,' flag has been set for processing for an ISC
*ST2 run. '/')
3 FORMAT(3X,'Inputs entered in ', A10,' will be converted to meters
```

```

*using '/3X,' a conversion factor of',F10.4, '. Output will be in
*meters.'/)
4  FORMAT(3X,'The UTMP variable is set to ',A4,'. The input is assum
*ed to be in'/4X,' UTM coordinates. BPIP will move the UTM origin
*to the first pair of'/4X,' UTM coordinates read. The UTM coordina
*tes of the new origin will '/4X,' be subtracted from all the other
* UTM coordinates entered to form '/4X,' this new local coordinate
*system.'/)
5  FORMAT(3X,'UTMP is set to ',A4,'. The input is assumed to be in a
* local'/3x,' X-Y coordinate system as opposed to a UTM coordinate
*system.'/3x,' True North is in the positive Y direction.'/)
9  FORMAT(/3X,'The ',A2,' flag has been set for processing for an ISC
*LT2 run.'/)
11  FORMAT(3X,'The new local coordinates will be displayed in parenthe
*ses just below'/4X,' the UTM coordinates they represent.'/)
12  FORMAT(/1X,A8,' has',I2,' tier(s) with a base elevation of', F8.2
*, ' ',A10)
13  FORMAT(47X,'(',F8.2,') meters',/)
14  FORMAT(
*' BUILDING TIER BLDG-TIER TIER NO. OF CORNER COORDINAT
*ES',
*/' NAME NUMBER NUMBER HEIGHT CORNERS X Y'
*/)
17  FORMAT(3X,'Plant north is set to',F7.2,' degrees with respect to T
*true North. '///)
18  FORMAT(4X,'The plant coordinates will appear as entered in the Sum
*mary output'/4x,'file and they will be adjusted to True North prio
*r to processing.'/4x,'The True North oriented coordinates appear b
*elow between'/4X,'the square brackets.')
47  FORMAT(' Caution: Blank spaces are not allowed in Stack names by
*ISC2 models.')
71  FORMAT (/' The number of stack-tier combinations entered, where e
*ach stack is at least 5L'' in from at least one of the edges of t
*heir respective tier roofs, is:',I4,/)
73  FORMAT(8x,' These stack/building-tiers are: '//
* ' STACK BUILDING - TIER'/
* ' NAME NUMBER NAME NO. NO.')
72  FORMAT( 1X, A8,2X, I4, 3x,A8,2X, 2I4)
1007  FORMAT(/' * Results are based on Determinants 1 & 2 on pages 1 &
* 2 of the GEP'/' Technical Support Document. Determinant 3 m
*ay be investigated for'/' additional stack height credit. Fin
*al values result after'/' Determinant 3 has been taken into co
*nsideration.'/
*' ** Results were derived from Equation 1 on page 6 of GEP Techni
*cal'/' Support Document. Values have been adjusted for any st
*ack-building'/' base elevation differences.'//
*' Note: Criteria for determining stack heights for modeling e
*mission'/

1025  FORMAT(10X,'*adjusted for a Stack-Building elevation difference of
*',F8.2)
2020  FORMAT(/' Summary By Direction Table'/
2022  FORMAT(' StkNo:', I3,' Stk Name:', A8, 23X,' Stack Ht:', F8.2)

to:

WRITE(*,*) 'The SWTN variable, ',SWTN,' is incorrectly ',
+ 'entered.'
STOP
WRITE(*,*) 'The UTMP variable, ',UTMP,' is incorrectly ',
+ 'entered.'
WRITE(*, *) 'THE NUMBER OF BUILDINGS ENTERED EXCEEDS THE ',
+ 'PARAMETER, MB'
WRITE(*, *) 'THE NUMBER OF TIERS/BUILDING EXCEEDS THE ',
* 'PARAMETER, MT'
WRITE(*, *) 'THE NUMBER OF TIERS/BUILDING EXCEEDS THE ',
* 'PARAMETER, MTS'
WRITE(*, *) 'THE NUMBER OF STACKS ENTERED EXCEEDS THE ',
+ 'PARAMETER, MKS'

2  FORMAT(/3X,'The ',A2,' flag has been set for processing for an ',
+'ISCST2 run.'/)

```

```

3  FORMAT(3X,'Inputs entered in ', A10,' will be converted to ',
+ 'meters using '/3X,' a conversion factor of',F10.4,
+ '. Output will be in meters.'/)
4  FORMAT(3X,'The UTMP variable is set to ',A4,'. The input is ',
+ 'assumed to be in'/4X,
+ ' UTM coordinates. BPIP will move the UTM origin',
+ ' to the first pair of'/4X,' UTM coordinates read.',
+ ' The UTM coordinates of the new origin will '/4X,
+ ' be subtracted from all the other UTM coordinates entered to ',
+ 'form '/4X,' this new local coordinate system.'/)
5  FORMAT(3X,'UTMP is set to ',A4,'. The input is assumed to be in',
+ ' a local'/3x,' X-Y coordinate system as opposed to a UTM',
+ ' coordinate system.'/3x,' True North is in the positive Y',
+ ' direction.'/)
9  FORMAT(/3X,'The ',A2,' flag has been set for processing for an',
+ ' ISCLT2 run.'/)
11  FORMAT(3X,'The new local coordinates will be displayed in parent',
+ 'heses just below'/4X,' the UTM coordinates they represent.',/)
12  FORMAT(/1X,A8,' has',I2,' tier(s) with a base elevation of',
+ ' F8.2,' ',A10)
13  FORMAT(47X,'(',F8.2,') meters',/)
14  FORMAT(' BUILDING TIER BLDG-TIER TIER NO. OF CORNER ',
+ 'COORDINATES',/
+ ' NAME NUMBER NUMBER HEIGHT CORNERS X Y'
+ '/')
17  FORMAT(3X,'Plant north is set to',F7.2,' degrees with respect to',
+ ' True North. '/)
18  FORMAT(4X,'The plant coordinates will appear as entered in the ',
+ 'Summary output'/4x,'file and they will be adjusted to True ',
+ 'North prior to processing.'
+ '/4x,'The True North oriented coordinates appear ',
+ 'below between'/4X,'the square brackets.')
47  FORMAT(' Caution: Blank spaces are not allowed in Stack names ',
+ 'by ISC2 models.')
71  FORMAT (/' The following lists the stacks that have been identi',
+ 'fied'/' as being atop the noted building-tiers.'/)
72  FORMAT(9X, ' STACK BUILDING TIER'/
+ ' 9X, ' NAME NO. NAME NO. NO.')
73  FORMAT(10X, A8, I4, 5X, A8, 2(1X, I5))
115  FORMAT(' Wind flow passing', I4,' degree direction.')
1007  FORMAT(/' * Results are based on Determinants 1 & 2 on pages 1',
+ '& 2 of the GEP'/' Technical Support Document. Determinan',
+ 't 3 may be investigated for'/' additional stack height cred',
+ 'it. Final values result after'/' Determinant 3 has been ta',
+ 'ken into consideration.'/
+ '** Results were derived from Equation 1 on page 6 of GEP Tech',
+ 'nical'/' Support Document. Values have been adjusted for a',
+ 'ny stack-building'/' base elevation differences.'//
+ ' Note: Criteria for determining stack heights for modeling',
+ ' emission'/'
1025  FORMAT(10X,'*adjusted for a Stack-Building elevation difference',
+ ' of',F8.2)

```

11. The roof stack algorithm was rewritten from:

```

READ (10,*) L5SQAT
WRITE (14,71) L5SQAT

IF(L5SQAT .GT. 0) THEN
WRITE(14,73)
END IF

DO 70 L = 1, L5SQAT
READ(10, *) S, I, J
C = (I-1) * MXTRS + J
GEPIN(S, C, C) = 1
LFLAT(S, C) = 1
WRITE(14,72) STKN(S), S, BTN(I), I, J
70  CONTINUE

```

to:

```

C      Detect if a stack is on top of a roof
C      where the stack could be > 5L from an upwind roof edge.

      L5SQAT = 0
      DO 70 LS = 1, NS
      DO 70 LB = 1, NB
      DO 70 LT = 1, NTRS(LB)
      C = (LB-1) * MXTRS + LT
      DIRTT = 0.0
      DO 74 LD = 1, ND(LB, LT)
      IF (YS(LS) .EQ. Y(LB,LT,LD)) THEN
      IF (XS(LS) .GT. X(LB,LT,LD)) THEN
      DIRT(LD) = 270.
      END IF
      IF (XS(LS) .LT. X(LB,LT,LD)) THEN
      DIRT(LD) = 90.
      END IF
      IF (XS(LS) .EQ. X(LB,LT,LD)) THEN
      GEPIN(LS,C,C,1) = 1
      LFLAT(LS,C) = 1
      DIRTT = 360.
      GOTO 77
      END IF
      ELSE
      XCOMP = X(LB,LT,LD) - XS(LS)
      YCOMP = Y(LB,LT,LD) - YS(LS)
      DIRT(LD) = DATAN(XCOMP/YCOMP) / DTR2
      IF (YCOMP .GT. 0.0) DIRT(LD) = 360. + DIRT(LD)
      IF (YCOMP .LE. 0.0) DIRT(LD) = 180. + DIRT(LD)
      IF (DIRT(LD) .GT. 360.) DIRT(LD) = DIRT(LD) - 360.
      END IF
74      CONTINUE
      DO 76 LD = 1, ND(LB,LT)
      LD1 = LD
      LD2 = LD + 1
      IF (LD2 .GT. ND(LB,LT)) LD2 = 1
      AA = DIRT(LD2) - DIRT(LD1)
      IF (AA .LE. -180.) AA = 360. + AA
      IF (AA .GT. 180.) AA = AA - 360.
      DIRTT = DIRTT + AA
76      CONTINUE
77      CONTINUE
      IF (ABS(360. - DIRTT) .LT. 0.02) THEN
      L5SQAT = L5SQAT + 1
      IF (L5SQAT .EQ. 1) THEN
      WRITE (14,71)
      WRITE (14,72)
      END IF
      GEPIN (LS,C,C,1) = 1
      LFLAT (LS,C) = 1
      WRITE(14,73) STKN(LS), LS, BTN(LB), LB, LT
      END IF
70      CONTINUE
      IF (L5SQAT .EQ. 0) THEN
      WRITE(14,*) ' '
      WRITE(14,*) ' No stacks have been detected as being atop',
      + ' any structures.'
      END IF

```

12. An extra argument was added to the CALL GPC statements to identify when the calculations were for a single tier (1) or group of tiers (2). The lines are not contiguous. The following lines were changed from:

```

      CALL GPC (D, I, C, S, TW, WS, HTA, C)

      CALL GPC (D, I, C1, S, TW, WS, HTA, TL1)

      CALL GPC (D, I, C1, S, TW, WS, HTA, TL1)

      CALL GPC (D, I, C1, S, TW, WS, HTA, TL1)

```

```

CALL GPC (D,I,C1,S,TW,WS, HTA, TL1)
CALL GPC (D, I, C1, S, TW, WS, HTA, TL1)
CALL MXBWH(D, I, S, C, TW, HTA, WS, CH)
CALL MXBWH(D, I, S, C, TW, HTA, WS, CH)

```

to:

```

CALL GPC (D, I, C, S, TW, WS, HTA, C, 1)
CALL GPC (D, I, C1, S, TW, WS, HTA, TL1,2)
CALL GPC (D, I, C1, S, TW, WS, HTA, TL1, 2)
CALL GPC (D, I, C1, S, TW, WS, HTA, TL1, 2)
CALL GPC (D,I,C1,S,TW,WS, HTA, TL1, 2)
CALL GPC (D, I, C1, S, TW, WS, HTA, TL1, 2)
CALL MXBWH(D, I, S, C, TW, HTA, WS, CH, 1)
CALL MXBWH(D, I, S, C, TW, HTA, WS, CH, 2)

```

13. Additional message are now written to the screen during program execution to help the user identify how far the program has progressed. The lines are not contiguous.

```

WRITE(*,*) ' Looking for and calculating',
+         ' any group of tiers GEP values'
WRITE(*,*) '   for a wind flow starting at 0.25 degrees.'

IF (MOD(D/4.,10.) .EQ. 0.) THEN
  WRITE(*,115) D/4
END IF

WRITE(*,*) ' Calculating single tier downwash values.'

WRITE(*,*) ' Calculating group of tiers downwash values.'

```

14. The line of code causing an additional stack-building elevation difference to be added to the Preliminary GEP Stack Height Value in the Output and Summary tables for values over 65 meters has been deleted. The two lines of code affected have been changed from:

```

GP = GEP(S) - DIF
PV = MAX (G65, GP)

```

to:

```

PV = MAX (G65, GEP(S))

```

15. The value TNUM2(C) was not initialized in the single tier algorithms. This contributed to the wrong group of tiers being used occasionally in the Summary file. The following lines of code were changed from:

```

CH = C
HTA = TH(I,J)

```

to:

```

CH = C
TNUM2(C) = 1
HTA = TH(I,J)

```

16. To facilitate a slight restructuring of the Summary file output, the Write statement with a Format number of 2026 was moved down several lines and changed to Format number 1026. This output is for a single tier.

```

WRITE(14,2022) S, STKN(S), SH(S)
WRITE(14,2026) MXPBH(S,D), MXPBW(S,D), MHWE(S,D)
WRITE(14,2027) GEPBH(S), GEPBW(S), GEP(S)
IF (MHWE(S,D) .EQ. 0.0) THEN
WRITE(14,*) '      No single tier affects this stack for this
&direction.'
ELSE
DIF = SB(S) - BELEV(MI(S))
WRITE(14,1025) DIF
WRITE(14,2028) MI(S), BTN(MI(S)), MJ(S)
END IF

```

to:

```

WRITE(14,2022) S, STKN(S), SH(S)
WRITE(14,2027) GEPBH(S), GEPBW(S), GEP(S)
IF (MHWE(S,D) .EQ. 0.0) THEN
WRITE(14,*) '      No single tier affects this stack for ',
+          'this direction.'
ELSE
WRITE(14,1026) MXPBH(S,D), MXPBW(S,D), MHWE(S,D)
DIF = SB(S) - BELEV(MI(S,1))
WRITE(14,1025) DIF
WRITE(14,2028) MI(S,1), BTN(MI(S,1)), MJ(S,1)
END IF

```

17. The 2026 Format statement was divided into two statements. The original statement was designed to print a set of directional maximum wake effect height values. The new statements are designed to print two sets of directional maximum wake effect height values. One set for a single tier for a particular direction and another set for a group of tiers for a particular direction. The original statement was changed from:

```

2026 FORMAT(5X, 'Directional MAX:  BH:',F7.2,'  PBW:',F7.2,
* ' *Wake Effect Ht:', F8.2)

```

to:

```

1026 FORMAT(5X, 'Single tier MAX:  BH:',F7.2,'  PBW:',F7.2,
* ' *Wake Effect Ht:', F8.2)
2026 FORMAT(3X,'Combined tier MAX:  BH:',F7.2,'  PBW:',F7.2,
* ' *Wake Effect Ht:', F8.2)

```

18. See statement #16 above.

from:

```

DO 2012 S = 1, NS
WRITE(14,2022) S, STKN(S), SH(S)
WRITE(14,2026) MXPBH(S,D), MXPBW(S,D), MHWE(S,D)
WRITE(14,2027) GEPBH(S), GEPBW(S), GEP(S)
IF (MTNUM(S) .LT. 2) THEN
WRITE(14,*) '      No combined tiers affect this stack for th
&is direction.'
ELSE
IF (DFLG(S,D) .EQ. 1) THEN
DIF = SB(S) - BELEV(MI(S))
WRITE(14,1025) DIF
WRITE(14,2023) MTNUM(S)
WRITE(14,2024) (MTLIST(S,M), M = 1, MTNUM(S))
ELSE
WRITE(14,*) '      No combined tiers affect this stack for
& this direction'
END IF

```

to:

```

DO 2012 S = 1, NS
WRITE(14,2022) S, STKN(S), SH(S)
WRITE(14,2027) GEPBH(S), GEPBW(S), GEP(S)
IF (MTNUM(S,2) .LT. 2) THEN

```

```

        WRITE(14,*) '          No combined tiers affect this stack for ',
+           'this direction.'
      ELSE
        IF (DFLG(S,D) .EQ. 1) THEN
          WRITE(14,2026) DPBH(S), DPBW(S), DHWE(S)
          DIF = SB(S) - BELEV(MI(S,2))
          WRITE(14,1025) DIF
          WRITE(14,2023) MTNUM(S,2)
          WRITE(14,2024) (MTLIST(S,M,2), M = 1, MTNUM(S,2))
        ELSE
+          WRITE(14,*) '          No combined tiers affect this stack ',
          'for this direction'
      END IF

```

19. A BPIP header has been added to the output file. The lines of code were changed from:

```

      IF (SWT .EQ. 0) THEN
        WRITE (12, 461) IMON, IDAY, IYR
        WRITE (12, 462) IHR, IMIN, ISEC
        WRITE (12, 297)

```

to:

```

      IF (SWT .EQ. 0) THEN
        WRITE (12, 461)
        WRITE (12, 462) IMON, IDAY, IYR
        WRITE (12, 463) IHR, IMIN, ISEC
        WRITE (12, 297)

```

20. The following non contiguous comment out lines were deleted:

```

C          WRITE (12,297)
C          IZ = D
C          IZ = D
C          WRITE (12,411) STKN (S)
C          WRITE (12,297)
C          WRITE (12,292) (MH(S,L) , L = 1, ND16)
C          WRITE (12,292) (MW(S,L) , L = 1, ND16)

```

21. In the Subroutine GPC, an argument was added to the subroutine arguments to pass a flag that differentiates between values for a single tier and values for a group of tiers. Other changes made to this subroutine are based on this new argument and the need to solidly separate the single tier results from the group of tier results. The lines below are not contiguous but they are all part of the Subroutine GPC. The lines were changed from:

```

SUBROUTINE GPC (D, I, C, S, TW, WS, HTA, CH)

COMMON /GP/ GEP(MSK), GEPBH(MSK), GEPBW(MSK), GEPIN(MSK,MBT,MBT)

COMMON /MIJ/ MI(MSK), MJ(MSK)

HWE = HTA + BELEV(I) - SB(S) + 1.5 * WS
GEPIN(S, C, CH) = 1
IF (HWE .GT. GEP(S)) THEN

  GTNUM(S) = TNUM2(C)
  MI(S) = I
  MJ(S) = C - (I-1)*MT
  GDIRS(S) = FLOAT(D)/4

```

to:

```

SUBROUTINE GPC (D, I, C, S, TW, WS, HTA, CH, IG)

COMMON /GP/ GEP(MSK), GEPBH(MSK), GEPBW(MSK),
* GEPIN(MSK,MBT,MBT,2)

```

```

COMMON /MIJ/ MI(MSK,2), MJ(MSK,2)

HWE = HTA + BELEV(I) - SB(S) + 1.5 * WS
GEPIN(S, C, CH, IG) = 1
IF (HWE .GT. GEP(S)) THEN

    GTNUM(S) = TNUM2(C)
    MI(S,1) = I
    MJ(S,1) = C - (I-1)*MT
    GDIRS(S) = FLOAT(D)/4

```

22. In the Subroutine MXBWH, an argument was added to the subroutine arguments to pass a flag that differentiates between values for a single tier and values for a group of tiers. Other changes made to this subroutine are based on this new argument and the need to solidly separate the single tier results from the group of tier results. The lines below are not contiguous but they are all part of the Subroutine MXBWH. The lines were changed from:

```

SUBROUTINE MXBWH(D, I, S, C, TW, HTA, WS, TL1)

COMMON /DE/ DE, DFLG(MSK, MD)
COMMON /GP/ GEP(MSK), GEPBH(MSK), GEPBW(MSK), GEPIN(MSK,MBT,MBT)
COMMON /MIJ/ MI(MSK), MJ(MSK)
COMMON /MTNM/ MTNUM(MSK), MTLIST(MSK,MBT)

IF (GEPIN(S,C,TL1) .EQ. 1) THEN

    DE = D
    DHWE = 0.0
    DPBH = 0.0
    DPBW = 0.0
    DO 577 SS = 1, MSK
577     DFLG(SS, D) = 0
    END IF

    IF (HWE - MHWE(S, D) .GT. .001) THEN

        DFLG(S,D) = 1
        MTNUM(S) = TNUM2(C)
        MI(S) = I
        MJ(S) = C - (I-1)*MT
        DO 578 M = 1, MTNUM(S)
578     MTLIST(S,M) = TLIST2(C,M)
        CONTINUE

    IF (TNUM2(C) .GT. 1) THEN
        IF (HWE .GT. DHWE) THEN
            MFLG = 2
        END IF
        IF ((ABS(HWE-DHWE).LT. .001) .AND. (PBW .LT. DPBW)) THEN
            MFLG = 2

            DFLG(S, D) = 1
            DHWE = HWE
            DPBW = PBW
            MFLG = 0
            MTNUM(S) = TNUM2(C)
            MI(S) = I
            MJ(S) = C - (I-1) * MT
            DO 579 M = 1, MTNUM(S)
579     MTLIST(S,M) = TLIST2(C,M)
            CONTINUE

to:

SUBROUTINE MXBWH(D, I, S, C, TW, HTA, WS, TL1, IG)

COMMON /DE/ DE, DFLG(MSK, MD), DHWE(MSK), DPBH(MSK), DPBW(MSK)
COMMON /GP/ GEP(MSK), GEPBH(MSK), GEPBW(MSK),
+ GEPIN(MSK,MBT,MBT,2)

```

```

COMMON /MIJ/ MI(MSK,2), MJ(MSK,2)
COMMON /MTNM/ MTNUM(MSK,2), MTLIST(MSK,MBT,2)

IF (GEPIN(S,C,TL1,IG) .EQ. 1) THEN

  DE = D
  DO 577 SS = 1, MSK
  DHWE(SS) = 0.0
  DPBH(SS) = 0.0
  DPBW(SS) = 0.0
  MTNUM(SS,IG) = 0
577  DFLG(SS, D) = 0
  END IF
  IF (HWE - MHWE(S, D) .GT. .001) THEN

    DFLG(S,D) = 1
    MTNUM(S,1) = TNUM2(C)
    MI(S,IG) = I
    MJ(S,IG) = C - (I-1)*MT
    DO 578 M = 1, MTNUM(S,IG)
    MTLIST(S,M,IG) = TLIST2(C,M)
578  CONTINUE

  IF (TNUM2(C) .GT. 1) THEN
    IF (HWE .GT. DHWE(S)) THEN
      MFLG = 2
    END IF
    IF ((ABS(HWE-DHWE(S)).LT. .001) .AND. (PBW .LT. DPBW(S))) THEN
      MFLG = 2
    END IF

    DFLG(S, D) = 1
    DHWE(S) = HWE
    DPBW(S) = PBW
    DPBH(S) = PBH
    MFLG = 0
    MTNUM(S,IG) = TNUM2(C)
    MI(S,IG) = I
    MJ(S,IG) = C - (I-1) * MT
    DO 579 M = 1, MTNUM(S,IG)
    MTLIST(S,M,IG) = TLIST2(C,M)
579  CONTINUE

```

#### USER'S GUIDE REVISIONS

Six pages of the BPIP user's guide were revised. These pages include the user's guide cover and pages 3-1, 3-3, 3-5 through 3-7. On the front cover, a revision date and a note stating that these revisions were made to the electronic copy only were added. On the other five pages, the Model Change Bulletin Julian date was added to the bottom right corner of each page.

Any edits made to the text of these pages involves the change from manually entered roof stack data to the automatic detection of which stacks are on which roofs. No pages were deleted or added. Therefore, a direct substitution of the revised pages with the respective original pages is possible without having to print the entire user's guide.

Model Change Bulletin

MCB#4

BPIP (dated 95086)

This is a formal notification of changes made to the Building Profile Input Program (BPIP) source code. The changes result from a reported problem with the roof stack algorithm. The changes are discussed first and are then followed by a list of changes made to the source code. The model is available for downloading under the Related Programs menu of the Air Quality Models area of SCRAM BBS.

Six pages to the user's guide were revised and discussed in the previous Model Change Bulletin. These pages are available in the current ZIP package, BPIP.ZIP, for replacement of original pages in the BPIP user's guide.

DISCUSSION OF SOURCE CODE CHANGES

A problem was found with the automated roof stack algorithm when tier corners are entered in a counterclockwise direction. Some of the roof stacks under certain conditions are not detected as being on a particular roof. An absolute (ABS) function was added to a line of code that corrects this problem.

SOURCE CODE CHANGES

The following is a list of changes made to BPIP:

1. The SCRAM header and format line have been changed to reflect the new BPIP version date. The header lines have been changed from:

```
C *          BUILDING PROFILE INPUT PROGRAM (DATED 95039)          *
C *
C *          *** SEE BPIP MODEL CHANGE BULLETIN MCB#3 ***          *
```

to:

```
C *          BUILDING PROFILE INPUT PROGRAM (DATED 95086)          *
C *
C *          *** SEE BPIP MODEL CHANGE BULLETIN MCB#4 ***          *
```

and the format line has been changed from:

```
461  FORMAT(30X,'BPIP (Dated: 95039)')
```

to:

```
461  FORMAT(30X,'BPIP (Dated: 95086)')
```

2. The following modification comment has been added:

```
C          March 27, 1995 - To fix a problem with the roof stack
C                          algorithm when tier coordinates are
C                          entered in a counterclock-wise
C                          direction. Some stacks were not
C                          detected as being on a roof.
```

3. An additional absolute (ABS) function was added to the following line of code so tier corners, when entered in a counterclockwise direction, would be properly identified as being or not being on a roof. The line of code was changed from:

```
IF (ABS(360. - DIRTT) .LT. 0.02) THEN
```

to:

```
IF (ABS(360. - ABS(DIRTT)) .LT. 0.02) THEN
```

