

EPA-454/R-93-038
(Revised April 21, 2004)
(Electronic copy only)

**USER'S GUIDE
TO THE
BUILDING PROFILE INPUT PROGRAM**

U.S. ENVIRONMENTAL PROTECTION AGENCY
Office of Air Quality Planning and Standards
Technical Support Division
Research Triangle Park, North Carolina 27711

October 1993

3.0 USER'S INSTRUCTIONS

The following subsections contain information on setting up and running BPIP. The BPIP program is straight forward in its operation.

3.1 INPUT PREPARATION

Before preparing the input data to BPIP, the structure-source relationships should be assessed with respect to the GEP and building downwash guidance. Any relationships that do not appear to be covered by the guidance or in some way appear ambiguous should be discussed with the Regional Modeling Contact. For example, a multi-tiered building with two separate towers can be treated in two ways. The first way as strictly a multitiered building where each tier is treated as a stand-alone structure. The second way as a multitiered building with two towers that may be combined. If the second way is recommended, the base tier would have to be entered twice; once with the first tower and once with the second tower.

Stacks can be on top of roofs and also be more than 5L downwind from an upwind roof edge. The main algorithms in BPIP were not designed to process these stacks if they are further than 5L downwind from a roof edge. This is contrary to guidance. An algorithm was written to automatically detect when a stack is on a roof.

Data preparation for use by the Huber-Synder downwash algorithm should be prepared as if it were to be used by the Schulman-Scire downwash algorithm. Input data to BPIP should use normal building dimensions and orientation in all cases. BPIP will calculate 36 pairs of BH and PBW values for input to the ISCST2 model or 16 pairs of input for the ISCLT2 model for each stack.

It is up to the user to determine the suitability of BPIP to meet the needs of the modeling situation. There are additional conditions and situations that are discussed in the GEP technical support document which are not fully programmed into BPIP. The conditions and situations concern the use of various formulae for determining GEP stack height based upon the date a stack came into existence. BPIP has not been programmed to use the 2.5H formula that is prescribed for stacks in existence prior to January 12, 1979 but after December 31, 1970. However, for low or squat buildings only, the 2.5H formula is equivalent to the $H + 1.5L$ formula for calculating GEP stack height. The user needs to read and understand the guidance and this guide before running BPIP.

If the structure and location values are not in meters, the conversion factor will be applied during the input process and

3.2 INITIAL PROGRAM SETTINGS

BPIP was originally programmed using PARAMETER statements to set the maximum size of several arrays. Thereafter, these array sizes could only be size adjusted by editing the PARAMETER values governing array sizes and then recompiling the program. Many users needed larger array sizes but some did not have a compiler available. A better method for sizing arrays was requested.

As a solution, ALLOCATABLE and ALLOCATE statements have been added to BPIP. The ALLOCATABLE statement identifies which fixed arrays are to become adjustable and specifies the number dimensions assigned to each ALLOCATABLE Array. In BPIP, the dimension sizes can be based on number of buildings, tiers, tier sides, stacks, etc. Once the maximum dimension sizes have been determined, the ALLOCATE statement is used to set the maximum dimension size(s) for each allocatable array. In BPIP, this readjustment is done by reading an input file and searching for the maximum size number for each dimension. After the maximum dimension sizes have been determined and allocated for each array, the input file is rewound to the beginning of the file and read again. The individual input values are read and then properly stored in the appropriately sized arrays. There is no need to recompile BPIP.

BPIP was rewritten to Fortran 90 standards and recompiled with Compaq Visual FORTRAN version 6.6 compiler. The OPEN and CLOSE statements have been commented out in the source code.

3.3 INPUT FORMAT

The input format consists of 12 different types of data input lines or records. Table 3-1 identifies each record type and its purpose. An example is given with each type of input.

The types of record in Table 3-1 are required to be entered into an ASCII file in a certain sequence. In Table 3-2 is an outline of that sequence. After the eighth line, entry is dependent upon the number of buildings, tiers, tier corners, and stacks. An annotated example of the sequence, using actual data, is shown in Table 3-3.

3.4 EXECUTION OF BPIP

Once the input file has been prepared and saved to disk, BPIP is ready to be executed. The execution line is as follows:

```
BPIP input_filename output_filename summary_filename
```

which can be executed from a Command Prompt or saved in a *.BAT file and double clicked from Windows Explorer. Normal runtime using a Pentium 4 2.0 Ghz is about 1 second for the smaller input files to about 4 seconds for the larger input files. The larger files consist of 3 combinable buildings with 3 tiers each and

Table 3-1 Cont'd

Record Type	<u>Description</u>
TIER _{i,j}	<p>Contains the number of tier corner coordinates to follow in the following record lines and the tier height with respect to the building base elevation. The number of corners is an integer while the base elevation value is real. Example line: 6 20.5</p>
CORNER _{i,j,k}	<p>Contains a pair of tier corner x-y coordinate values if the GRID line contains a 'UTMN' flag. If the GRID line contains a 'UTMY' flag, the coordinate pair are treated as UTM Easting and Northing coordinates, respectively. Example line: -23.5 46.75</p>
S-NO	<p>Contains the integer value of the number of stacks to be processed. Example line: 9</p>
STACK _s	<p>Contains the stack data which consists of: Stack name - up to 8 characters allowed with no spaces allowed in the name. The name has to be between single quotes. Stack base elevation - real value Stack height - real value measured from the stack base. Stack coordinates - see CORNER for format Example line: 'Stk3' 123.1 10.5 -45.3 -34.1</p>

Table 3-2

Data Input Sequence

<u>Record Type</u>	<u>Placement</u>
TITLE	First line of file
PROCESS	Second line of file
UNITS	Third line of file
GRID	Fourth line of file
B-NO	Five line of file
BLDG 1	Sixth line of file
TIER 1,1	Seventh line of file
CORNER 1,1,1	Eight line of file
CORNER 1,1,2	Etc.
.	
CORNER 1,1,m	
TIER 1,2	
. CORNER 1,2,1	
.	
. CORNER 1,2,n	
TIER 1,t	
CORNER 1,t,1	
.	
CORNER 1,t,r	
BLDG 2	
.	(repeat of the TIER-CORNER pattern for BLDG 1.)
.	
BLDG b	(followed by a repeat of the BLDG 1 pattern.)
S-NO	
STACK 1	
.	
STACK s	

Table 3-3

Input Example

Record Type	Example input
TITLE	'Simple building'
PROCESS	'LT'
UNITS	'Feet' 0.3048
GRID	'UTMY' 270.0
B-NO	2
BLDG 1	'L - Bldg' 1 10.0
TIER 1,1	6 20.0
CORNER 1,1,1	-10.3 -20.5
CORNER 1,1,2	-10.3 80.1
CORNER 1,1,3	40.3 80.1
CORNER 1,1,4	40.3 30.3
CORNER 1,1,5	90.4 30.3
CORNER 1,1,6	90.4 -20.5
BLDG 2	'Proc. 1' 1 15.5
TIER 2,1	4 10.0
CORNER 2,1,1	110.3 120.5
CORNER 2,1,2	110.3 180.1
CORNER 2,1,3	140.3 180.1
CORNER 2,1,4	140.3 120.5
S-NO	4
STACK 1	'Stack100' 5. 25.0 -10.0 -20.0
STACK 2	'Stack101' 5. 25.0 10.0 15.0
STACK 3	'Stack102' 5. 25.0 136.0 121.0
STACK 4	'Stack103' 5. 25.0 118.0 103.0