INPUT MACROS AND GENERAL LIBRARY FEATURES

Input Macros

The "Input Macros" feature was added to the Building Description Language in DOE-2.1D to increase the flexibility of BDL. This feature is intended for advanced users who are already familiar with preparing BDL input. The basic capabilities are:

(1) Incorporating external files containing pieces of BDL into the main BDL input stream. This is also called the "General Library" feature.

(2) Selectively accepting or skipping portions of the input.

(3) Defining a block of input with parameters and later referencing this block.

(4) Performing arithmetic and logical operations on the input.

(5) Input macro debugging and listing control.

These capabilities are invoked in BDL by using *macro commands.* Macro commands are preceded by ## to distinguish them from regular BDL commands. After execution by the BDL processor, macro commands produce regular lines of BDL input that are shown in the BDL echo print. Following are descriptions of the macro commands associated with the above capabilities. A detailed example of input macros is given at the end of this section. The user should look at this example before reading the macro command descriptions. This example is also in the *Sample Run Book (2.1E)* as "Parameterized Building".

(1) Incorporating External Files

##include {includefilename}

This command puts all of the lines in an external file into the BDL input stream starting right after the command line. The name of the file that is included is the concatenation of {prefixpathname}, entered using **##fileprefix**, and {includefilename}. The lines in the external file will be listed in the BDL echo so that the user can see exactly what is being included. When all the lines in the external file have been read in, input reverts back to the original input file at the line following the **##include** command.

##fileprefix {prefixpathname}

specifies a pathname that will be prefixed to the filename given in an **##include** command. The **##fileprefix** command allows commonly-used include files to be kept in a directory other than the directory in which the current DOE-2 input file resides.

Example: on VAX/VMS, the combination

##fileprefix DRC2:[GUEST.LIBRARY]

##include SCHEDULES.INP

will include into the BDL stream the file whose full name is

DRC2:[GUEST.LIBRARY]SCHEDULES.INP

##includesilent {includefilename}

This command is identical to **##include**, except that the lines in the included file will not be listed in the BDL echo.

##nosilent

Overrides the listing suppression of **##includesilent**. Used for debugging purposes only. After **##nosilent**, all following **##includesilent** commands are treated as **##include** commands.

Example: Assume the following files contain the indicated lines:

Main input file: input1.inp	External file: file2.inp
line 1a	line 2a
##include file2.inp	line 2b
line 1b	line 2c
line 1c	

The end result of processing **##include** input1.inp will be:

```
line 1a     (from input1.inp)
line 2a        (from file2.inp)
line 2b        (from file2.inp)
line 2c        (from file2.inp)
line 1b     (from input1.inp)
line 1c     (from input1.inp)
```

External files can also contain **##include** commands, as shown in the following example:

Main input file: input1.inp	First external file: file2.inp	Second external file: file3.inp
line 1a	line 2a	line 3a
##include file2.inp	line 2b	line 3b
line 1b	**##include** file3.inp	line 3c
line 1c	line 2c	line 3d

The end result of processing **##include** inputl.tmp will be:

```
line 1a     (from input1.inp)
line 2a       (from file2.inp)
line 2b       (from file2.inp)
line 3a          (from file3.inp)
line 3b          (from file3.inp)
line 3c          (from file3.inp)
line 3d          (from file3.inp)
line 2c       (from file2.inp)
line 1b     (from input1.inp)
line 1c     (from input1.inp)
```

Note: Up to nine **##include** commands can be nested. However, there should be no recursion. This is an example of a recursion:

```
file1.inp contains ##include file2.inp
file2.inp contains ##include file1.inp
```

(2) Selectively Accepting or Skipping Lines of Input

The **##if** series of commands is used to selectively accept or skip lines of input according to the following sequence:

```
##if {condition1}
        line1a
        line1b
        ....

##elseif {condition2}
        line2a
        line2b
        ....

##elseif {condition3}
        line3a
        line3b
        ....

##else
        lineNa
        lineNb
        ....

##endif
```

Then the lines that will be included into the BDL stream are:

line1a line1b	if {condition 1} is TRUE,
	otherwise
line2a line2b	if {condition 2} is TRUE,
	otherwise
line3a line3b	if {condition 3} is TRUE,
	otherwise
line*N*a line*N*b	if {condition 1}, {condition 2}, {condition 3} are all FALSE.

There are six different **##if...** commands:

Command			**Result**
##ifdef	{macro name}	:	if macro name defined, include following lines
##ifndef	{macro name}	:	if macro name NOT defined, include following lines
##if	{condition}	:	if condition is TRUE, include following lines
##elseif	{condition}	:	if condition is TRUE, and previous conditions are FALSE, include following lines
##else		:	if all previous conditions are FALSE, include following lines
##endif		:	indicates the end of the if block

Notes:

{macro name} is explained in (3), below.

{condition} is 0 or BLANK meaning FALSE, and any other character meaning TRUE.

##ifdef and **##ifndef** do not have corresponding **##elseif** commands, but they do have corresponding **##else** and **##endif** commands.

(3) Defining Blocks of Input

The **##def** command allows a block of input text to be defined and given a name. The block of text can then be inserted anywhere in the BDL stream by simply referencing the name of the block. (This process is called "macro expansion".) The block can have parameters (also called arguments) that can be given different values each time the block is referenced.

The syntax of the **##def** command is as follows:

```
            unique name
                 |             zero or more arguments                __ macro text
                 |                  |                                |
                 |                  |                                |
                 ↓                  ↓                                ↓
      ##def     macro-name       [  arg1   arg2,arg3      ...]      text line 1
    ↑         ↑                 ↑  ↑      ↑      ↑       ↑        ↑
    |_        |                 |  |      |_     |      _|        |
     |        |                 |  |         |   |    |           |__ one
     |        |                 |  |         |   |    |               or
zero        one            zero____|       space(s)                   more
or          or             or              or comma                   spaces
more        more           more
spaces      spaces         spaces
```

Example: Define a macro with name "all_ones":

```
##def  all_ones    SCHEDULE    THRU DEC 31
                               (ALL) (1,24) (1.0)

##enddef
```

Then, in the BDL input stream, when we say :

```
SCHED1=   all_ones[]   ..
                    ↑
                    |_____ the square braces
                           are required
```

the result is equivalent to:

```
SCHED1 = SCHEDULE    THRU DEC 31
                     (ALL) (1,24) (1.0) ..
```

Macro definitions may have one or more arguments; the maximum number of arguments is 32. When a macro with arguments is referenced, its arguments must be given values.

Example: Define a macro with name "sched" and argument "x":

```
##def sched[x]    SCHEDULE    THRU DEC 31 (ALL) (1,24) (x)

##enddef
```

Then, when we put the following in the BDL input stream

```
SCHED2 = sched[.20] ..
SCHED3 = sched[.33] ..
```

the result is equivalent to:

```
SCHED2 = SCHEDULE  THRU DEC 31 (ALL) (1,24) (.20) ..
SCHED3 = SCHEDULE  THRU DEC 31 (ALL) (1,24) (.33) ..
```

Macro names must be unique (except see **##set1** below); i.e., when a macro name is defined it cannot be defined again.

To summarize, commands you use to define macros are the following:

##def macro-name **[arg1,..,arg*n*]** macro-text

Defines a macro with the name "macro-name" and arguments "arg1" through "arg*n*". "Macro-text" is one or more lines of text. If there are no arguments, the syntax is ##def macro-name macro-text

##enddef

Indicates the end of the macro definition initiated by **##def.**

##def1 macro-name **[arg1,..,arg*n*]** macro-text

This is the same as **##def** but there is only one line of text so that the terminating command **##enddef** is not required.

##set1 macro-name macro-text

Like **##def1** but has no arguments and macro-text is evaluated before storing. "Macro-text is evaluated" means that if macro-text contains other macros, these macros will be expanded, and the expanded text becomes the macro-text defined by **##set1**.

Example:
```
##def1   xx   123
##set1   yy   xx[ ]
```

is equivalent to:
```
##set1   yy   123
```

##set1 can also be used to redefine macro-name.

Example:
```
##set1   x    0
  .
  .
  .
##set1   x    #eval[x[ ]+1]
```
(see (4), Arithmetic Operations, for description of the #eval macro.)

(4) Arithmetic Operations

The built-in macro called #**eval**[ ] can be used to perform arithmetic, literal, and logical operations. It can be abbreviated to #[ ].

#eval[X OP Y] or #[X OP Y] gives the result X OP Y. The allowed values for X, OP, and Y, and the corresponding result, are shown in the following table.

X*	OP**	Y	Result
number	+ (plus)	number	number
number	− (minus)	number	number
number	* (times)	number	number
number	/ (divided by)	number	number
number	min	number	number
number	max	number	number
number	mod	number	number
number	**	number	number
SIN	OF	number (degrees)	number
COS	OF	number (degrees)	number
TAN	OF	number (degrees)	number
SQRT	OF	number	number
ABS	OF	number	number
ASIN	OF	number	number (degrees)
ACOS	OF	number	number (degrees)
ATAN	OF	number	number
INT	OF	number	number
LOG10	OF	number	number
LOG	OF	number	number
literal1	// (concatenate)	literal2	literal "literal1literal2"
literal1	/// (concatenate)	literal2	literal "literal1 literal2"
literal	EQS (=)	literal	logical (true or false)
literal	NES (≠)	literal	logical (true or false)
logical	AND	logical	logical (true or false)
logical	OR	logical	logical (true or false)
	NOT	logical	logical (true or false)
number	EQ (=)	number	logical (true or false)
number	NE (≠)	number	logical (true or false)
number	GT (>)	number	logical (true or false)
number	GE (≥)	number	logical (true or false)
number	LT (<)	number	logical (true or false)
number	LE (≤)	number	logical (true or false)

* Upper or lower case is allowed for SIN, COS, etc.

** Upper or lower case is allowed for OF, EQS, etc.

Example

```
#eval[ 1 + 2 ]    when expanded becomes 3.
#eval[ 1 + #eval[ 2 * 3 ] ]    when expanded becomes 7.
```

Example

```
#set1 city Washington
    TITLE  LINE-1  #["large office" /// city[ ] ]
    gives
    TITLE  LINE-1  "large office  Washington"
```

The following example illustrates the use of **#eval** inside **#if** commands:

```
##if  #[ city[ ] EQS "Chicago" ]

##if  #[#[ city[ ] EQS "Chicago" ] and #[ occup[ ] NES "low"] ]
```

Notes:

1. For logical values:

   False = 0 or BLANK,
   True = any other character

2. A literal must be enclosed inside a pair of double quotes if it contains BLANKs or reserved characters like [] ( ) , * =

   E.g., "abc *def"

   Otherwise, the quotes around the literals are optional.

3. Literal concatenation operators // and /// produce quoted literals.

   E.g., #[large /// office] gives "large office"

4. Literals are case sensitive. For example, "Chicago", "CHICAGO" "chicago" are distinct.

(5) Macro Debugging and Listing Control

##list

Turn on listing; echo of input lines on the OUTPUT file is enabled. This is the default condition.

##nolist

Turn off listing; echo of input lines on the output file is disabled.

##show

Start printing expanded line on output file. After this command, if a macro expansion was done, the expanded line is printed on the output file. In this way you can see the end result of macro expansions, which is the input as seen by the BDL processor.

##noshow

Stop printing expanded line on output file. This is the default condition.

##showdetail

Start printing each macro expansion. After this command, every time a macro expansion is done the result of the expansion is printed. This can produce lots of output.

##noshowdetail

Stop printing each macro expansion. This is the default condition.

##traceback

Give full traceback when printing an error message. After this command, if there is a BDL error, a full traceback of the macro expansions in progress is printed. This is the default condition.

##notraceback

Don't give full traceback when printing an error message.

##write

Start writing expanded text into file 22. This is similar to **##show** except that the expanded lines are written into file 22. Therefore, file 22 will contain only the text that will be seen by the BDL processor. This file is used only for debugging purposes. It allows you to see what the macro-processed input file looks like.

##nowrite

Stop writing expanded text into file 22. This is the default condition.

##symboltable

Prints table of current macro names. All of the macro names that are defined will be printed.

##clear

Clear all macro definitions. All the macro names defined up to this point will be deleted.

##reserve TEXT *k* NAMES *l* STACK *m*

Allocates memory.
Reserves *k* words of space in AA array for macro definition storage.
Reserves *l* positions in macro definition names table.
Reserves *m* words of stack space.

If used, the **##reserve** command must precede all other macro commands in the BDL input. This command should be used only if one or more of the following error messages is received:

"Need more memory for storing macro definitions"
Use "**##reserve** TEXT *nnnnnn*" command to get more memory. Current value of

nnnnnn is: _ _ _

"Macro table capacity exceeded"
Use "**##reserve** NAMES *nnnnnn*" command to get more memory. Current value of *nnnnnn* is: _ _ _

"Macro stack overflow"
Use "**##reserve** STACK *nnnnnn*" command to get more memory. Current value of *nnnnnn* is: _ _ _

##$ <**comment**>
Allows you to enter comment lines inside a macro. <**comment**> is printed in the BDL echo but is not acted on by the macro processor.

Example:
This example shows the use of the **##set**, **##include**, **##eval** and **##if** commands. Let an external file called cities.lib contain the following text:

```
##if    #[city[ ] EQS  CHICAGO]

    BUILDING-LOCATION    LAT=41.88
                         LON=87.63
                         ALT=600
                         T-Z=6     $Chicago$
                         ..


##elseif    #[city[ ] EQS  WASHINGTON]

    BUILDING-LOCATION    LAT=38.9
                         LON=77
                         ALT=50
                         T-Z=5     $Washington$
                         ..

##else
    ERROR--City Undefined

##endif
```

Then the BDL input

```
        INPUT  LOADS ..
        ##set1        city CHICAGO
        ##include     cities.lib
             .
             .
             .
```

will be converted, after macro processing, to:

```
        INPUT  LOADS ..
        BUILDING-LOCATION    LAT=41.88
                             LON=87.63
                             ALT=600
                             T-Z=6     $Chicago$
                             ..
             .
             .
             .
```

Listing Format

The format of listing in DOE-2.1D has been changed to give information about the status of the input macros, as shown in the following:

```
 1.1.1 * 123 * ...... DOE-2 input line
 - - - -------  ---------------------------------
 ↑ ↑ ↑    ↑      ↑
 | | |    |      |
 | | |    |    echo of DOE-2 input line
 | | |    |
 | | |  line number ( if the current line is being skipped
 | | |                by "##if..." etc, this is indicated
 | | |                by printing "- 123 -" instead of
 | | |                "* 123 *" in the line number field;
 | | |                if the current line is part of a
 | | |                macro command, it is indicated by
 | | |                printing "# 123 #" )
 | | |
 | |macro expansion nesting level
 | |
 |"##if" nesting level
 |
"##include" nesting level
```

Library Example

Following is a listing of individual files. File samp7.inp is the DOE-2 input file; the library files that are needed by samp7.inp are the following: samp7loc.inp and samp7lib.inc.

DOE-2 input and "include" files

```
$----- file : samp7.inp ----------- DOE2.1E sample run 7 ----------------------

##write

input LOADS ..
##includesilent samp7loc.inc

   run-period  JAN 1 1981  THRU  JAN 7 1981  ..

   LOCATION[chicago]  azimuth = 0 ..

WA-1-2 =  layers   material = (WD01,PW03,IN02,GP01)  ..
RB-1-1 =  layers   material = (RG01,BR01,IN22,WD01)  i-f-r = .76  ..
WALL-typ1 =  construction     layers = WA-1-2  ..
ROOF-typ1 =  construction     layers = RB-1-1  ..
IWF-typ1  =  construction     u = 0.5   ..
FLOOR-typ1=  construction     u = 0.05  ..
GT-typ1   =  glass-type       p=1  s-c=.60 ..

$-----SPACE CONDITIONS-----$

OCC-SCH=          schedule  THRU DEC 31  (WD) (1,7)(0) (8,17)(1) (18,24)(0)
                                         (WEH)(1,24)(0) ..
CORE-LITE-SCH=    schedule  THRU DEC 31  (WD) (1,6)(0) (7,18)(1) (19,24)(0)
                                         (WEH)(1,24)(0) ..
INF-SCH=          schedule  THRU DEC 31  (WD) (1,7)(0) (8,17)(1) (18,24)(0)
                                         (WEH)(1,24)(0) ..
PERIM-LITE-SCH=   schedule  THRU DEC 31  (WD) (1,6)(0) (7,18)(.8) (19,24)(0)
                                         (WEH)(1,24)(0) ..

CORE=    space-conditions   t = (75)
                            people-schedule = OCC-SCH  people-heat-gain = 450
                                area/person = 100
                            lighting-schedule = CORE-LITE-SCH
                                lighting-w/sqft = 2
                                light-to-space = .75  light-to-return = .25
                            inf-method = AIR-CHANGE   inf-schedule = INF-SCH
                                air-changes/hr = 0.8
                            z-type = CONDITIONED      ..
PERIM=   space-conditions   like CORE
                            area/person = 90
                            lighting-schedule = PERIM-LITE-SCH
                            air-changes/hr = 0.6  ..
```

```
$------ set some macros for glass type and constructions.

##setl  GLASS_TYPE  GT-typl
##setl  EW_CONS     WALL-typl
##setl  IW_CONS     IWF-typl
##setl  ROOF_CONS   ROOF-typl
##setl  UGF_CONS    FLOOR-typl

##include  samp7lib.inc

##$---- now generate each floor.
##$
##$    #   W    D    H   perim_D  window       space conditions for
##$                               frac    core  north  south  east   west
##$   -- ---  ---  ---  -------  -----   -----  -----  -----  -----  -----
$
$----- ground floor
FLOOR[ g, 100,  50,  10,    15   , .20,    CORE, PERIM, PERIM, PERIM, PERIM ]
$----- 1st    floor
FLOOR[ 1, 100,  50,  10,    15   , .30,    CORE, PERIM, PERIM, PERIM, PERIM ]
$----- top floor
FLOOR[ t, 100,  50,  10,    15   , .30,    CORE, PERIM, PERIM, PERIM, PERIM ]

end ..
compute LOADS ..
stop ..
```

```
##$  file : samp7loc.inc
##$
##$      This file contains the LOCATION[Name] macro, that produces the
##$        'building-location' command.  Here 'Name' is the name of the
##$        city.
##$      An example of usage is :
##$                                    LOCATION[chicago]  azimuth = 0 ..
##$
##def  LOCATION[Name]
##nolist
  building-location
##if      #[Name eqs boston]
                lat = 42.37   lon = 71.07    alt = 50     t-z = 5
##elseif #[Name eqs newyork]
                lat = 40.72   lon = 74.00    alt = 50     t-z = 5
##elseif #[Name eqs philadelphia]
                lat = 39.95   lon = 75.17    alt = 50     t-z = 5
##elseif #[Name eqs detroit]
                lat = 42.33   lon = 83.00    alt = 600    t-z = 5
##elseif #[Name eqs chicago]
                lat = 41.88   lon = 87.63    alt = 600    t-z = 6
```

```
##elseif #[Name eqs sanfrancisco]
               lat = 37.78  lon = 122.42  alt = 50    t-z = 8
##elseif #[Name eqs losangeles]
               lat = 34.07  lon = 118.25  alt = 50    t-z = 8
##elseif #[Name eqs sandiego]
               lat = 32.72  lon = 117.15  alt = 50    t-z = 8
##elseif #[Name eqs phoenix]
               lat = 33.45  lon = 112.07  alt = 1000  t-z = 7
##else
Abort -- building location undefined in file : samp7loc.inc
##endif

##$ defaults for all locations:
      hol = YES  daylight-savings = YES
##list
##enddef
```

```
##$ file : samp7lib.inc
##$
##$      This file contains the FLOOR[...] and PERIM-SPACE[...] macros.
##$      The FLOOR[...] macro is used for specifying one floor of the
##$          building with given dimensions.  It uses the PERIM-SPACE[...]
##$          macro to create the perimeter spaces.
##$
##$-----------------------------------------------------------------
##$ The following macro defines one floor of the building.
##$     Its arguments are :
##$         floor_NUM : g for ground floor; 1 for first floor ;
##$                     2 for second floor; etc. ; t for top floor .
##$                     s for single floor building.
##$         floor_W   : floor width.
##$         floor_D   : floor depth.
##$         floor_H   : floor height.
##$         floor_perim_D : depth of the perimeter spaces.
##$         floor_winFrac : window to wall ratio for exterior surfaces.
##$         cond_C     : u-name of space conditions for the interior space.
##$         cond_N     : u-name of space conditions for the north space.
##$ In addition, the following macros must be set to define glass type,
##$     and wall constructions:
##$                    GLASS_TYPE  glass type of windows.
##$                    EW_CONS     exterior-wall construction.
##$                    IW_CONS     interior-wall construction.
##$                    ROOF_CONS   roof construction.
##$                    UGF_CONS    underground-wall,floor construction.
##$                    WINDOW_HEIGHT .window height. ( default = 3.0 )
##$-----------------------------------------------------------------
```

```
##$
##def FLOOR[floor_NUM,floor_W,floor_D,floor_H,floor_perim_D,floor_winFrac
           ,cond_C, cond_N, cond_S, cond_E, cond_W]
  ##$
  ##$                                  --->|     |<--- floor_perim_D
  ##$
  ##$
  ##$          ----------wwwwwwwwwwwwww----------
  ##$          |\                              /|         ^
  ##$          | \             _N             / |         |
  ##$          |   \                        /   |         |
  ##$          |     -------------------          |         |
  ##$          w      |                   |       w         |
  ##$          w      |                   |       w         |
  ##$          w  _W  |         _C        |  _E   w     floor_D
  ##$          w      |                   |       w         |
  ##$          w      |                   |       w         |
  ##$          |     -------------------          |         |
  ##$          |    /                      \     |         |
  ##$          |  /            _S            \   |         |
  ##$          |/                              \ |         v
  ##$          ----------wwwwwwwwwwwwww----------
  ##$
  ##$
  ##$          <--------      floor_W ------->
##$
  ##ifndef  WINDOW_HEIGHT
    ##setl  WINDOW_HEIGHT  3.0
  ##endif
##$
  ##ifndef  Z-SP
    ##setl  Z-SP  0
  ##endif
##$
  ##setl  FLOOR-HEIGHT  floor_H
  ##setl  FLOOR-NUM     floor_NUM
  ##setl  FLOOR-PERIM-D floor_perim_D
##$
  ##setl  SP-NAM0  #["SP_" // floor_NUM]
##$
  ##setl  AREA-SP  #[ floor_perim_D * #[ floor_W - floor_perim_D ] ]
  #[SP-NAM0[] // "_N"] = PERIM-SPACE[ _N, floor_W, floor_D, Z-SP[],180
                                     , AREA-SP[], floor_W, floor_winFrac
                                     , cond_N ]

  #[SP-NAM0[] // "_S"] = PERIM-SPACE[ _S, 0, 0, Z-SP[],  0
                                     , AREA-SP[], floor_W, floor_winFrac
                                     , cond_S ]

  ##setl  AREA-SP  #[floor_perim_D * #[ floor_D - floor_perim_D ] ]
```

```
#[SP-NAM0[] // "_E"] =  PERIM-SPACE[ _E, floor_W, 0, Z-SP[], -90
                                   , AREA-SP[], floor_D, floor_winFrac
                                   , cond_E ]
    ##set1  AREA-IW  #[ 1.4142 * #[ floor_perim_D * floor_H ] ]
    ##if  #[AREA-IW[] GT 0 ]
                     interior-wall  area = AREA-IW[]
                                    next-to = #[ SP-NAM0[] // "_N" ]
                                    construction = IW_CONS[]  ..
                     interior-wall  area = AREA-IW[]
                                    next-to = #[ SP-NAM0[] // "_S" ]
                                    construction = IW_CONS[]  ..
    ##endif

#[SP-NAM0[] // "_W"] =  PERIM-SPACE[ _W, 0, floor_D, Z-SP[], 90
                                   , AREA-SP[], floor_D, floor_winFrac
                                   , cond_W ]
    ##if  #[AREA-IW[] GT 0 ]
                     interior-wall  area = AREA-IW[]
                                    next-to = #[ SP-NAM0[] // "_N" ]
                                    construction = IW_CONS[]  ..
                     interior-wall  area = AREA-IW[]
                                    next-to = #[ SP-NAM0[] // "_S" ]
                                    construction = IW_CONS[]  ..
    ##endif

$-------- Core space --------$
##set1  TMP1  #[ 2 * floor_perim_D ]
##set1  AREA-SP  #[ #[floor_W - TMP1[]] * #[floor_D - TMP1[]] ]
##if #[ AREA-SP[] GT 0 ]
    #[SP-NAM0[] // "_C"] =  space  x = floor_perim_D  y = floor_perim_D
                                   z = Z-SP[]  azimuth = 0  area = AREA-SP[]
                                   volume = #[ AREA-SP[] * FLOOR-HEIGHT[] ]
                                   space-conditions=cond_C  ..
        interior-wall  area = #[ #[floor_W - TMP1[]] * floor_H ]
                       next-to = #[ SP-NAM0[] // "_S" ]
                       construction = IW_CONS[]  ..
        interior-wall  area = #[ #[floor_W - TMP1[]] * floor_H ]
                       next-to = #[ SP-NAM0[] // "_N" ]
                       construction = IW_CONS[]  ..
        interior-wall  area = #[ #[floor_D - TMP1[]] * floor_H ]
                       next-to = #[ SP-NAM0[] // "_E" ]
                       construction = IW_CONS[]  ..
        interior-wall  area = #[ #[floor_D - TMP1[]] * floor_H ]
                       next-to = #[ SP-NAM0[] // "_W" ]
                       construction = IW_CONS[]  ..

        ##if #[ floor_NUM EQS "g" ]
          underground-floor  area = AREA-SP[]  construction = UGF_CONS[] ..fR
        ##else
          interior-wall      area = AREA-SP[]
```

```
                              next-to = #[ #["SP_" // FLOOR-PREV[]] // "_C"]
                              construction = IW_CONS[]  ..
          ##endif

          ##if #[ floor_NUM EQS "t" ]
            roof          x = floor_perim_D  y = floor_perim_D
                          h = #[floor_D - #[ 2 * floor_perim_D ] ]
                          w = #[floor_W - #[ 2 * floor_perim_D ] ]
                          construction = ROOF_CONS[]  ..
          ##endif
   ##endif
   ##$
   ##$ update the z-coordinate.
   ##setl  Z-SP  #[ Z-SP[] + FLOOR-HEIGHT[] ]
   ##setl  FLOOR-PREV  floor_NUM
   ##$
##enddef

##$
##$-------------------------------------------------------------------
##$ The following macro defines one perimeter space.
##$      Its arguments are :
##$          sname :        part of the name of space.  ( N, S, E, W )
##$          xx,yy,zz :     x, y, z  location of the space.
##$          sazim :        space azimuth.
##$          sarea :        space area.
##$          swidth :       space width.
##$          swinFrac :     window to wall ratio of exterior surfaces.
##$          space_cond : u-name of space conditions.
##$ This macro uses the FLOOR-NUM[], FLOOR-HEIGHT[] and WINDOW_HEIGHT[]
##$      that are set by the FLOOR macro.
##$ It also uses :
##$      GLASS_TYPE macro to set the glass-type of windows.
##$      EW_CONS     macro to set the exterior-wall construction.
##$      IW_CONS     macro to set the interior-wall construction.
##$      ROOF_CONS   macro to set the roof construction.
##$      UGF_CONS    macro to set the underground-wall,floor construction.
##$-------------------------------------------------------------------
##$
##def PERIM-SPACE[sname,xx,yy,zz,sazim,sarea,swidth,swinFrac,space_cond]

    space              x = xx  y = yy  z = zz  azimuth = sazim  area = sarea
                       volume = #[ sarea * FLOOR-HEIGHT[] ]
                       space-conditions = space_cond  ..
        exterior-wall  x = 0  y = 0  azimuth = 0  h = FLOOR-HEIGHT[]
                       w = swidth  construction = EW_CONS[]  ..
  ##setl TMP1 #[ #[swinFrac * #[FLOOR-HEIGHT[] * swidth]] / WINDOW_HEIGHT[]]
  ##if #[ TMP1[] GT 0 ]
          window       x = #[ #[swidth - TMP1[]] / 2 ]
                       y = 3.0
```

```
                      w = TMP1[]   h = WINDOW_HEIGHT[]
                      glass-type = GLASS_TYPE[]  ..
  ##endif

  ##if #[ #[FLOOR-NUM[] EQS "g"] or #[FLOOR-NUM[] EQS "s"] ]
        underground-floor  area = sarea  construction = UGF_CONS[] ..
  ##else
        interior-wall    area = sarea  construction = IW_CONS[]
                         next-to = #[ #["SP_" // FLOOR-PREV[]] // sname ] ..
  ##endif

  ##if #[ #[FLOOR-NUM[] EQS "t"] or #[FLOOR-NUM[] EQS "s"] ]
        roof  x = 0  y = 0
              h = FLOOR-PERIM-D[]
              w = #[ swidth - FLOOR-PERIM-D[] ]
              construction = ROOF_CONS[] ..
  ##endif
##$
##enddef
```

Following is the BDL listing that results from running the samp7.inp file as an input to DOE-2.

```
     ****   ***   *****        ***      *   ****
     *  *   *  *  *           *   *    **   *
     *  *   *  *  *****  ***     *      *   ***
     *  *   *  *  *            **       *   *
     ****   ***   *****       ***** *  ***  ****


 B U I L D I N G    E N E R G Y    A N A L Y S I S    P R O G R A M


                          DEVELOPED BY
      LAWRENCE BERKELEY LABORATORY/UNIVERSITY OF CALIFORNIA,
    WITH THE ASSISTANCE OF HIRSCH & ASSOCIATES, CAMARILLO, CA
                      WITH MAJOR SUPPORT FROM
                  UNITED STATES DEPARTMENT OF ENERGY
    ASSISTANT SECRETARY FOR CONSERVATION AND RENEWABLE ENERGY
                  OFFICE OF BUILDING TECHNOLOGIES
               BUILDING SYSTEMS AND MATERIALS DIVISION
                    AND ADDITIONAL SUPPORT FROM
    SOUTHERN CALIFORNIA EDISON CO., PACIFIC GAS AND ELECTRIC CO.,
      GAS RESEARCH INSTITUTE, ELECTRIC POWER RESEARCH INSTITUTE
      COPYRIGHT 1993 REGENTS OF THE UNIVERSITY OF CALIFORNIA,
                   LAWRENCE BERKELEY LABORATORY


******************** L E G A L   N O T I C E ******************
* This program was prepared as an account of work sponsored by the *
* United States government.  Neither the United States nor the     *
* Department of Energy, nor any of their employees, nor any of     *
* their contractors, subcontractors, or their employees makes any  *
* warranty, express or implied, or assumes any legal liability or  *
* responsibility for the accuracy, completeness or usefulness of   *
* any information, apparatus, product or process disclosed, or     *
* represents that its use would not infringe privately owned rights.*
********************************************************************

          LBL RELEASE SEP 1993        version : 2.1E-B20
```

```
  *    1 * $----- file : samp7.inp ---------- DOE2.1E sample run 7 --------------------
  *    2 *
  #    3 # ##write
  *    4 *
  *    5 * input  LOADS ..

            L D L   P R O C E S S O R   I N P U T   D A T A

                  Thu Jul 29 16:39:33 1993LDL RUN  1

  #    6 # ##includesilent  samp7loc.inc
file : INPUT2.TMP
  *    7 *
  *    8 *      run-period  JAN 1 1981  THRU  JAN 7 1981  ..
  *    9 *
  *   10 *      LOCATION[chicago]  azimuth = 0 ..
  .1#  36 # ##list
    *  11 *
    *  12 * WA-1-2 =  layers   material = (WD01,PW03,IN02,GP01)  ..
    *  13 * RB-1-1 =  layers   material = (RG01,BR01,IN22,WD01)  i-f-r = .76  ..
    *  14 * WALL-typ1 =  construction       layers = WA-1-2  ..
    *  15 * ROOF-typ1 =  construction       layers = RB-1-1  ..
    *  16 * IWF-typ1  =  construction       u = 0.5    ..
    *  17 * FLOOR-typ1=  construction       u = 0.05   ..
    *  18 * GT-typ1   =  glass-type         p=1  s-c=.60 ..
    *  19 *
    *  20 * $-----SPACE CONDITIONS-----$
    *  21 *
    *  22 * OCC-SCH=          schedule  THRU DEC 31 (WD) (1,7)(0) (8,17)(1) (18,24)(0)
    *  23 *                                         (WEH)(1,24)(0) ..
    *  24 * CORE-LITE-SCH=  schedule  THRU DEC 31 (WD) (1,6)(0) (7,18)(1) (19,24)(0)
    *  25 *                                         (WEH)(1,24)(0) ..
    *  26 * INF-SCH=          schedule  THRU DEC 31 (WD) (1,7)(0) (8,17)(1) (18,24)(0)
    *  27 *                                         (WEH)(1,24)(0) ..
    *  28 * PERIM-LITE-SCH= schedule  THRU DEC 31 (WD) (1,6)(0) (7,18)(.8) (19,24)(0)
    *  29 *                                         (WEH)(1,24)(0) ..
    *  30 *
    *  31 * CORE=   space-conditions  t = (75)
    *  32 *                           people-schedule = OCC-SCH  people-heat-gain = 450
    *  33 *                               area/person = 100
    *  34 *                           lighting-schedule = CORE-LITE-SCH
    *  35 *                               lighting-w/sqft = 2
    *  36 *                               light-to-space = .75  light-to-return = .25
    *  37 *                           inf-method = AIR-CHANGE   inf-schedule = INF-SCH
    *  38 *                               air-changes/hr = 0.8
    *  39 *                           z-type = CONDITIONED         ..
    *  40 * PERIM=  space-conditions  like CORE
    *  41 *                           area/person = 90
    *  42 *                           lighting-schedule = PERIM-LITE-SCH
    *  43 *                           air-changes/hr = 0.6  ..
```

```
     *  44 *
     *  45 * $------ set some macros for glass type and constructions.
     *  46 *
     #  47 # ##set1  GLASS_TYPE  GT-typ1
     #  48 # ##set1  EW_CONS     WALL-typ1
     #  49 # ##set1  IW_CONS     IWF-typ1
     #  50 # ##set1  ROOF_CONS   ROOF-typ1
     #  51 # ##set1  UGF_CONS    FLOOR-typ1
     *  52 *
     #  53 # ##include  samp7lib.inc
1    #   1 # ##$ file : samp7lib.inc
1    #   2 # ##$
1    #   3 # ##$       This file contains the FLOOR[... and PERIM-SPACE[... macros.
1    #   4 # ##$       The FLOOR[...] macro is used for specifying one floor of the
1    #   5 # ##$           building with given dimensions.  It uses the PERIM-SPACE[...]
1    #   6 # ##$           macro to create the perimeter spaces.
1    #   7 # ##$
1    #   8 # ##$---------------------------------------------------------------------
1    #   9 # ##$ The following macro defines one floor of the building.
1    #  10 # ##$     Its arguments are :
1    #  11 # ##$         floor_NUM : g for ground floor; 1 for first floor ;
1    #  12 # ##$                     2 for second floor; etc. ; t for top floor .
1    #  13 # ##$                     s for single floor building.
1    #  14 # ##$         floor_W   : floor width.
1    #  15 # ##$         floor_D   : floor depth.
1    #  16 # ##$         floor_H   : floor height.
1    #  17 # ##$         floor_perim_D : depth of the perimeter spaces.
1    #  18 # ##$         floor_winFrac : window to wall ratio for exterior surfaces.
1    #  19 # ##$         cond_C    : u-name of space conditions for the interior space.
1    #  20 # ##$         cond_N    : u-name of space conditions for the north space.
1    #  21 # ##$ In addition, the following macros must be set to define glass type,
1    #  22 # ##$     and wall constructions:
1    #  23 # ##$                   GLASS_TYPE  glass type of windows.
1    #  24 # ##$                   EW_CONS     exterior-wall construction.
1    #  25 # ##$                   IW_CONS     interior-wall construction.
1    #  26 # ##$                   ROOF_CONS   roof construction.
1    #  27 # ##$                   UGF_CONS    underground-wall,floor construction.
1    #  28 # ##$                   WINDOW_HEIGHT  window height. ( default = 3.0 )
1    #  29 # ##$---------------------------------------------------------------------
```

```
1    #  30 # ##$
1    #  31 # ##def FLOOR[floor_NUM,floor_W,floor_D,floor_H,floor_perim_D,floor_winFrac
1    #  32 #             ,cond_C, cond_N, cond_S, cond_E, cond_W]
1    #  33 #    ##$
1    #  34 #    ##$                          --->|      |<--- floor_perim_D
1    #  35 #    ##$
1    #  36 #    ##$
1    #  37 #    ##$       ----------wwwwwwwwwwwwww----------
1    #  38 #    ##$       |\                              /|        ^
1    #  39 #    ##$       | \              _N           /  |        |
1    #  40 #    ##$       |   \                       /    |        |
1    #  41 #    ##$       |      --------------------      |        |
1    #  42 #    ##$       w      |                  |      w        |
1    #  43 #    ##$       w      |                  |      w        |
1    #  44 #    ##$       w  _W  |         _C       |  _E  w      floor_D
1    #  45 #    ##$       w      |                  |      w        |
1    #  46 #    ##$       w      |                  |      w        |
1    #  47 #    ##$       |      --------------------      |        |
1    #  48 #    ##$       |    /                      \    |        |
1    #  49 #    ##$       |  /             _S           \  |        |
1    #  50 #    ##$       |/                              \|        v
1    #  51 #    ##$       ----------wwwwwwwwwwwwww----------
1    #  52 #    ##$
1    #  53 #    ##$
1    #  54 #    ##$       <--------       floor_W ------->
1    #  55 # ##$
1    #  56 #    ##ifndef  WINDOW_HEIGHT
1    #  57 #       ##setl  WINDOW_HEIGHT  3.0
1    #  58 #    ##endif
1    #  59 # ##$
1    #  60 #    ##ifndef  Z-SP
1    #  61 #       ##setl  Z-SP  0
1    #  62 #    ##endif
1    #  63 # ##$
1    #  64 #    ##setl  FLOOR-HEIGHT  floor_H
                        .
                        .
                        .

                 Example Shortened
                        .
                        .
                        .
1    # 196 #    ##endif
1    # 197 # ##$
1    # 198 # ##enddef
file : INPUT2.TMP
     *  54 *
     #  55 # ##$---- now generate each floor.
     #  56 # ##$
     #  57 # ##$    #   W    D    H   perim_D  window      space conditions for
     #   58 # ##$                                        frac    core  north  south  east   west
     #   59 # ##$   -- ---  ---  ---   -------   -----  -----  -----  -----  -----  -----
     *   60 * $
     *   61 * $----- ground floor
     *   62 * FLOOR[ g, 100,  50,  10,    15  , .20,    CORE, PERIM, PERIM, PERIM, PERIM
   .1#   32 #    ##$
   .1#   33 #    ##$
   .1#   34 #    ##$                                 --->|     |<--- 15
   .1#   35 #    ##$
   .1#   36 #    ##$
   .1#   37 #    ##$          ----------wwwwwwwwwwwwww----------
   .1#   38 #    ##$          |\                               /|      ^
   .1#   39 #    ##$          | \            _N              /  |      |
   .1#   40 #    ##$          |  \                          /   |      |
   .1#   41 #    ##$          |   ---------------------         |      |
   .1#   42 #    ##$          w    |                   |        w      |
   .1#   43 #    ##$          w    |                   |        w      |
   .1#   44 #    ##$          w _W |        _C         |  _E    w     50
   .1#   45 #    ##$          w    |                   |        w      |
   .1#   46 #    ##$          w    |                   |        w      |
   .1#   47 #    ##$          |   ---------------------         |      |
   .1#   48 #    ##$          |   /                     \       |      |
   .1#   49 #    ##$          |  /           _S          \      |      |
   .1#   50 #    ##$          |/                          \     |      v
   .1#   51 #    ##$          ----------wwwwwwwwwwwwww----------
   .1#   52 #    ##$
   .1#   53 #    ##$          <--------      100       ------->
   .1#   54 # ##$
 .1.1#   55 #    ##ifndef  WINDOW_HEIGHT
 .1.1#   56 #       ##setl  WINDOW_HEIGHT  3.0
   .1#   57 #    ##endif
   .1#   58 # ##$
 .1.1#   59 #    ##ifndef  Z-SP
 .1.1#   60 #       ##setl  Z-SP   0
   .1#   61 #    ##endif
   .1#   62 # ##$
   .1#   63 #    ##setl   FLOOR-HEIGHT  10
   .1#   64 #    ##setl   FLOOR-NUM       g
   .1#   65 #    ##setl   FLOOR-PERIM-D 15
   .1#   66 # ##$
   .1#   67 #    ##setl   SP-NAM0  #["SP_" // g]
   .1#   68 # ##$
   .1#   69 #    ##setl   AREA-SP  #[ 15 * #[ 100 - 15 ] ]
   .1*   70 *    #[SP-NAM0[] // "_N"] =  PERIM-SPACE[ _N, 100, 50, Z-SP[],180
   .1*   71 *                                          , AREA-SP[], 100, .20
   .1*   72 *                                          , PERIM ]
   .2*  170 *
   .2*  171 *      space   x = 100   y = 50   z = Z-SP[]   azimuth = 180   area = AREA-S
   .2*  171 * P[]
   .2*  172 *                      volume = #[ AREA-SP[] * FLOOR-HEIGHT[] ]
```

```
  .2* 173 *                             space-conditions = PERIM  ..
  .2* 174 *            exterior-wall    x = 0  y = 0  azimuth = 0  h = FLOOR-HEIGHT[]
  .2* 175 *                             w = 100  construction = EW_CONS[]  ..
  .2# 176 #   ##set1 TMP1 #[ #[.20 * #[FLOOR-HEIGHT[] * 100]] / WINDOW_HEIGHT[]]
  .2# 177 #   ##if #[ TMP1[] GT 0 ]
.1.2* 178 *              window         x = #[ #[100 - TMP1[]] / 2 ]
.1.2* 179 *                             y = 3.0
.1.2* 180 *                             w = TMP1[]  h = WINDOW_HEIGHT[]
.1.2* 181 *                             glass-type = GLASS_TYPE[]  ..
  .2# 182 #   ##endif
  .2* 183 *
  .2# 184 #   ##if #[ #[FLOOR-NUM[] EQS "g"] or #[FLOOR-NUM[] EQS "s"] ]
.1.2* 185 *            underground-floor  area = AREA-SP[]  construction = UGF_CONS[] ..
  .2# 186 #   ##else
.1.2- 187 -            interior-wall      area = AREA-SP[]  construction = IW_CONS[]
.1.2- 188 -                               next-to = #[ #["SP_" // FLOOR-PREV[]] // _N ]  ..
  .2# 189 #   ##endif
  .2* 190 *
  .2# 191 #   ##if #[ #[FLOOR-NUM[] EQS "t"] or #[FLOOR-NUM[] EQS "s"] ]
.1.2- 192 -            roof  x = 0  y = 0
.1.2- 193 -                  h = FLOOR-PERIM-D[]
.1.2- 194 -                  w = #[ 100 - FLOOR-PERIM-D[] ]
.1.2- 195 -                  construction = ROOF_CONS[] ..
  .2# 196 #   ##endif
  .2# 197 # ##$
  .1*  73 *
```

.

.

.

Example Shortened

.

.

.

```
     *   65 * $----- top floor
     *   66 * FLOOR[ t, 100,  50,  10,    15  , .30,    CORE, PERIM, PERIM, PERIM, PERIM
   .1#  32 #    ##$
   .1#  33 #    ##$
   .1#  34 #    ##$                                        --->|       |<--- 15
   .1#  35 #    ##$
   .1#  36 #    ##$                         ----------wwwwwwwwwwwwww---------
   .1#  37 #    ##$                         |\                             /|        ^
   .1#  38 #    ##$                         | \             _N           /  |        |
   .1#  39 #    ##$                         |   \                      /    |        |
   .1#  40 #    ##$                         |       ------------------      |        |
   .1#  41 #    ##$                         w      |                  |     w        |
   .1#  42 #    ##$                         w      |                  |     w        |
   .1#  43 #    ##$                         w  _W  |        _C        | _E  w     50
   .1#  44 #    ##$                         w      |                  |     w        |
   .1#  45 #    ##$                         w      |                  |     w        |
   .1#  46 #    ##$                         |       ------------------      |        |
   .1#  47 #    ##$                         |    /                     \    |        |
   .1#  48 #    ##$                         |  /            _S           \  |        |
   .1#  49 #    ##$                         |/                            \ |        v
   .1#  50 #    ##$                         ----------wwwwwwwwwwwwww---------
   .1#  51 #    ##$
   .1#  52 #    ##$
   .1#  53 #    ##$                         <--------      100      ------->
   .1#  54 # ##$
 .1.1#  55 #   ##ifndef  WINDOW_HEIGHT
 .1.1-  56 -     ##setl  WINDOW_HEIGHT  3.0
   .1#  57 #   ##endif
   .1#  58 # ##$
 .1.1#  59 #   ##ifndef  Z-SP
 .1.1-  60 -     ##setl  Z-SP  0
   .1#  61 #   ##endif
   .1#  62 # ##$
   .1#  63 #   ##setl  FLOOR-HEIGHT  10
   .1#  64 #   ##setl  FLOOR-NUM     t
   .1#  65 #   ##setl  FLOOR-PERIM-D 15
   .1#  66 # ##$
   .1#  67 #   ##setl  SP-NAM0  #["SP_" // t]
   .1#  68 # ##$
   .1#  69 #   ##setl  AREA-SP  #[ 15 * #[ 100 - 15 ] ]
   .1*  70 *   #[SP-NAM0[] // "_N"] =  PERIM-SPACE[ _N, 100, 50, Z-SP[],180
   .1*  71 *                                        , AREA-SP[], 100, .30
   .1*  72 *                                        , PERIM ]
   .2* 170 *
   .2* 171 *      space    x = 100   y = 50   z = Z-SP[]   azimuth = 180   area = AREA-S
   .2* 171 * P[]
   .2* 172 *                           volume = #[ AREA-SP[] * FLOOR-HEIGHT[] ]
   .2* 173 *                           space-conditions = PERIM  ..
   .2* 174 *          exterior-wall    x = 0   y = 0   azimuth = 0   h = FLOOR-HEIGHT[]
   .2* 175 *                           w = 100   construction = EW_CONS[]  ..
   .2# 176 #   ##set1 TMP1 #[ #[.30 * #[FLOOR-HEIGHT[] * 100]] / WINDOW_HEIGHT[]]
   .2# 177 #   ##if #[ TMP1[] GT 0 ]
 .1.2* 178 *          window       x = #[ #[100 - TMP1[]] / 2 ]
 .1.2* 179 *                       y = 3.0
 .1.2* 180 *                       w = TMP1[]  h = WINDOW_HEIGHT[]
 .1.2* 181 *                       glass-type = GLASS_TYPE[]  ..
   .2# 182 #   ##endif
   .2* 183 *
   .2# 184 #   ##if #[ #[FLOOR-NUM[] EQS "g"] or #[FLOOR-NUM[] EQS "s"] ]
 .1.2- 185 -        underground-floor  area = AREA-SP[]  construction = UGF_CONS[] ..
   .2# 186 #   ##else
 .1.2* 187 *        interior-wall     area = AREA-SP[]  construction = IW_CONS[]
 .1.2* 188 *                          next-to = #[ #["SP_" // FLOOR-PREV[]] // _N ]  ..
   .2# 189 #   ##endif
   .2* 190 *
   .2# 191 #   ##if #[ #[FLOOR-NUM[] EQS "t"] or #[FLOOR-NUM[] EQS "s"] ]
 .1.2* 192 *        roof  x = 0  y = 0
 .1.2* 193 *              h = FLOOR-PERIM-D[]
 .1.2* 194 *              w = #[ 100 - FLOOR-PERIM-D[] ]
 .1.2* 195 *              construction = ROOF_CONS[] ..
   .2# 196 #   ##endif
   .2# 197 # ##$
   .1*  73 *
   .1*  74 *   #[SP-NAM0[] // "_S"] =  PERIM-SPACE[ _S, 0, 0, Z-SP[],  0
   .1*  75 *                                      , AREA-SP[], 100, .30
   .1*  76 *                                      , PERIM ]
                     .
                     .
                     .
             Example Shortened
                     .
                     .
                     .
     *  67 *
     *  68 * end ..
     *  69 * compute LOADS ..
     *  70 * stop ..
```

==

Following is the listing of the file for022.dat. This file is requested by the "**##write**" command in file office.inp. Note that this file shows the end result of macro processing; i.e., all the macro commands are taken out and all macro expansions are done.

```
<   4 >
<   5 > input  LOADS ..
<   7 >
<   8 >      run-period  JAN 1 1981  THRU  JAN 7 1981  ..
<   9 >
<  10 >
<  11 >   building-location
<  21 >                    lat = 41.88   lon = 87.63   alt = 600   t-z = 6
<  33 >
<  35 >        hol = YES  daylight-savings = YES
<  11 >
<  12 > WA-1-2 =  layers   material = (WD01,PW03,IN02,GP01)  ..
<  13 > RB-1-1 =  layers   material = (RG01,BR01,IN22,WD01)  i-f-r = .76  ..
<  14 > WALL-typ1 =  construction       layers = WA-1-2  ..
<  15 > ROOF-typ1 =  construction       layers = RB-1-1  ..
<  16 > IWF-typ1  =  construction       u = 0.5   ..
<  17 > FLOOR-typ1=  construction       u = 0.05  ..
<  18 > GT-typ1   =  glass-type         p=1  s-c=.60 ..
<  19 >
<  20 > $-----SPACE CONDITIONS-----$
<  21 >
<  22 > OCC-SCH=          schedule  THRU DEC 31  (WD) (1,7)(0) (8,17)(1) (18,24)(0)
<  23 >                                          (WEH)(1,24)(0) ..
<  24 > CORE-LITE-SCH=    schedule  THRU DEC 31  (WD) (1,6)(0) (7,18)(1) (19,24)(0)
<  25 >                                          (WEH)(1,24)(0) ..
<  26 > INF-SCH=          schedule  THRU DEC 31  (WD) (1,7)(0) (8,17)(1) (18,24)(0)
<  27 >                                          (WEH)(1,24)(0) ..
<  28 > PERIM-LITE-SCH=   schedule  THRU DEC 31  (WD) (1,6)(0) (7,18)(.8) (19,24)(0)
<  29 >                                          (WEH)(1,24)(0) ..
<  30 >
<  31 > CORE=   space-conditions  t = (75)
<  32 >                           people-schedule = OCC-SCH  people-heat-gain = 450
<  33 >                               area/person = 100
<  34 >                           lighting-schedule = CORE-LITE-SCH
<  35 >                               lighting-w/sqft = 2
<  36 >                               light-to-space = .75  light-to-return = .25
<  37 >                           inf-method = AIR-CHANGE   inf-schedule = INF-SCH
<  38 >                               air-changes/hr = 0.8
<  39 >                           z-type = CONDITIONED       ..
<  40 > PERIM=  space-conditions  like CORE
<  41 >                           area/person = 90
<  42 >                           lighting-schedule = PERIM-LITE-SCH
<  43 >                           air-changes/hr = 0.6  ..
<  44 >
<  45 > $------ set some macros for glass type and constructions.
<  46 >
<  52 >
< 147 >
<  54 >
<  60 > $
<  61 > $----- ground floor
<  70 >   "SP_g_N" =
< 170 >
< 171 >     space                x = 100   y = 50   z = 0   azimuth = 180   area = 1275
< 172 >                          volume =  12750
< 173 >                          space-conditions = PERIM  ..
< 174 >         exterior-wall    x = 0   y = 0   azimuth = 0   h = 10
< 175 >                          w = 100   construction = WALL-typ1   ..
< 178 >           window         x =  16.666666031
< 179 >                          y = 3.0
< 180 >                          w = 66.666664124   h = 3.0
< 181 >                          glass-type = GT-typ1   ..
< 183 >
< 185 >         underground-floor   area = 1275   construction = FLOOR-typ1 ..
< 190 >
<  73 >
<  74 >   "SP_g_S" =
< 170 >
< 171 >     space                x = 0   y = 0   z = 0   azimuth = 0   area = 1275
< 172 >                          volume =  12750
< 173 >                          space-conditions = PERIM  ..
< 174 >         exterior-wall    x = 0   y = 0   azimuth = 0   h = 10
< 175 >                          w = 100   construction = WALL-typ1   ..
< 178 >           window         x =  16.666666031
< 179 >                          y = 3.0
< 180 >                          w = 66.666664124   h = 3.0
< 181 >                          glass-type = GT-typ1   ..
< 183 >
< 185 >         underground-floor   area = 1275   construction = FLOOR-typ1 ..
< 190 >
<  77 >
<  79 >   "SP_g_E" =
< 170 >
< 171 >     space                x = 100   y = 0   z = 0   azimuth = -90   area = 525
< 172 >                          volume =  5250
< 173 >                          space-conditions = PERIM  ..
< 174 >         exterior-wall    x = 0   y = 0   azimuth = 0   h = 10
< 175 >                          w = 50   construction = WALL-typ1   ..
< 178 >           window         x =  8.333333969
< 179 >                          y = 3.0
< 180 >                          w = 33.333332062   h = 3.0
< 181 >                          glass-type = GT-typ1   ..
< 183 >
< 185 >         underground-floor   area = 525   construction = FLOOR-typ1 ..
< 190 >
<  84 >                          interior-wall  area = 212.129989624
<  85 >                                         next-to = "SP_g_N"
<  86 >                                         construction = IWF-typ1  ..
<  87 >                          interior-wall  area = 212.129989624
<  88 >                                         next-to = "SP_g_S"
<  89 >                                         construction = IWF-typ1  ..
<  91 >
<  92 >  "SP_g_W" =
< 170 >
< 171 >    space               x = 0  y = 50  z = 0  azimuth = 90  area = 525
< 172 >                        volume =  5250
< 173 >                        space-conditions = PERIM  ..
< 174 >        exterior-wall  x = 0  y = 0  azimuth = 0  h = 10
< 175 >                       w = 50  construction = WALL-typ1  ..
< 178 >          window       x =  8.333333969
< 179 >                       y = 3.0
< 180 >                       w = 33.333332062  h = 3.0
< 181 >                       glass-type = GT-typ1  ..
< 183 >
< 185 >        underground-floor  area = 525  construction = FLOOR-typ1 ..
< 190 >
<  96 >                          interior-wall  area = 212.129989624
<  97 >                                         next-to = "SP_g_N"
<  98 >                                         construction = IWF-typ1  ..
<  99 >                          interior-wall  area = 212.129989624
< 100 >                                         next-to = "SP_g_S"
< 101 >                                         construction = IWF-typ1  ..
< 103 >
< 104 >  $-------- Core space --------$
< 108 >      "SP_g_C" =  space  x = 15  y = 15
< 109 >                                 z = 0  azimuth = 0  area = 1400
< 110 >                                 volume =  14000
< 111 >                                 space-conditions=CORE  ..
< 112 >          interior-wall  area =  700
< 113 >                         next-to = "SP_g_S".
< 114 >                         construction = IWF-typ1  ..
< 115 >          interior-wall  area =  700
< 116 >                         next-to = "SP_g_N"
< 117 >                         construction = IWF-typ1  ..
< 118 >          interior-wall  area =  200
< 119 >                         next-to = "SP_g_E"
< 120 >                         construction = IWF-typ1  ..
< 121 >          interior-wall  area =  200
< 122 >                         next-to = "SP_g_W"
< 123 >                         construction = IWF-typ1  ..
< 124 >
< 126 >          underground-floor  area = 1400  construction = FLOOR-typ1 ..
< 132 >
<  63 > $----- 1st   floor
<  70 >   "SP_1_N" =
< 170 >
< 171 >    space              x = 100  y = 50  z = 10  azimuth = 180  area = 1275
< 172 >                       volume =  12750
< 173 >                       space-conditions = PERIM  ..
< 174 >       exterior-wall   x = 0  y = 0  azimuth = 0  h = 10
< 175 >                       w = 100  construction = WALL-typ1  ..
< 178 >         window        x =  0
< 179 >                       y = 3.0
< 180 >                       w = 100  h = 3.0
< 181 >                       glass-type = GT-typ1  ..
< 183 >
< 187 >       interior-wall     area = 1275  construction = IWF-typ1
< 188 >                         next-to =  "SP_g_N"  ..
< 190 >
<  73 >
<  74 >  "SP_1_S" =
< 170 >
< 171 >    space              x = 0  y = 0  z = 10  azimuth = 0  area = 1275
< 172 >                       volume =  12750
< 173 >                       space-conditions = PERIM  ..
< 174 >       exterior-wall   x = 0  y = 0  azimuth = 0  h = 10
< 175 >                       w = 100  construction = WALL-typ1  ..
< 178 >         window        x =  0
< 179 >                       y = 3.0
< 180 >                       w = 100  h = 3.0
< 181 >                       glass-type = GT-typ1  ..
< 183 >
< 187 >       interior-wall     area = 1275  construction = IWF-typ1
< 188 >                         next-to =  "SP_g_S"  ..
< 190 >
<  77 >
<  79 >  "SP_1_E" =
< 170 >
< 171 >    space              x = 100  y = 0  z = 10  azimuth = -90  area = 525
< 172 >                       volume =  5250
< 173 >                       space-conditions = PERIM  ..
< 174 >       exterior-wall   x = 0  y = 0  azimuth = 0  h = 10
< 175 >                       w = 50  construction = WALL-typ1  ..
< 178 >         window        x =  0
< 179 >                       y = 3.0
< 180 >                       w = 50  h = 3.0
< 181 >                       glass-type = GT-typ1  ..
< 183 >
< 187 >       interior-wall     area = 525  construction = IWF-typ1
< 188 >                         next-to =  "SP_g_E"  ..
< 190 >
<  84 >                        interior-wall  area = 212.129989624
<  85 >                                       next-to = "SP_1_N"
<  86 >                                       construction = IWF-typ1  ..
<  87 >                        interior-wall  area = 212.129989624
<  88 >                                        next-to = "SP_1_S"
<  89 >                                        construction = IWF-typ1  ..
<  91 >
<  92 >  "SP_1_W" =
< 170 >
< 171 >     space          x = 0  y = 50  z = 10  azimuth = 90  area = 525
< 172 >                    volume =  5250
< 173 >                    space-conditions = PERIM  ..
< 174 >        exterior-wall  x = 0  y = 0  azimuth = 0  h = 10
< 175 >                    w = 50  construction = WALL-typ1  ..
< 178 >          window    x =  0
< 179 >                    y = 3.0
< 180 >                    w = 50  h = 3.0
< 181 >                    glass-type = GT-typ1  ..
< 183 >
< 187 >        interior-wall   area = 525  construction = IWF-typ1
< 188 >                        next-to =  "SP_g_W"  ..
< 190 >
<  96 >                      interior-wall  area = 212.129989624
<  97 >                                     next-to = "SP_1_N"
<  98 >                                     construction = IWF-typ1  ..
<  99 >                      interior-wall  area = 212.129989624
< 100 >                                     next-to = "SP_1_S"
< 101 >                                     construction = IWF-typ1  ..
< 103 >
< 104 >  $-------- Core space --------$
< 108 >      "SP_1_C" =  space  x = 15  y = 15
< 109 >                                   z = 10  azimuth = 0  area = 1400
< 110 >                                   volume =  14000
< 111 >                                   space-conditions=CORE  ..
< 112 >          interior-wall  area =  700
< 113 >                         next-to = "SP_1_S"
< 114 >                         construction = IWF-typ1  ..
< 115 >          interior-wall  area =  700
< 116 >                         next-to = "SP_1_N"
< 117 >                         construction = IWF-typ1  ..
< 118 >          interior-wall  area =  200
< 119 >                         next-to = "SP_1_E"
< 120 >                         construction = IWF-typ1  ..
< 121 >          interior-wall  area =  200
< 122 >                         next-to = "SP_1_W"
< 123 >                         construction = IWF-typ1  ..
< 124 >
< 128 >            interior-wall      area = 1400
< 129 >                               next-to =  "SP_g_C"
< 130 >                               construction = IWF-typ1  ..
< 132 >
<  65 > $----- top floor
<  70 >   "SP_t_N" =
< 170 >
< 171 >     space                x = 100  y = 50  z = 20  azimuth = 180  area = 1275
< 172 >                          volume =  12750
< 173 >                          space-conditions = PERIM  ..
< 174 >        exterior-wall     x = 0  y = 0  azimuth = 0  h = 10
< 175 >                          w = 100  construction = WALL-typ1  ..
< 178 >          window          x =  0
< 179 >                          y = 3.0
< 180 >                          w = 100  h = 3.0
< 181 >                          glass-type = GT-typ1  ..
< 183 >
< 187 >        interior-wall       area = 1275  construction = IWF-typ1
< 188 >                            next-to =  "SP_1_N"  ..
< 190 >
< 192 >        roof  x = 0  y = 0
< 193 >              h = 15
< 194 >              w =  85
< 195 >              construction = ROOF-typ1 ..
<  73 >
<  74 >   "SP_t_S" =
< 170 >
< 171 >     space                x = 0  y = 0  z = 20  azimuth = 0  area = 1275
< 172 >                          volume =  12750
< 173 >                          space-conditions = PERIM  ..
< 174 >        exterior-wall     x = 0  y = 0  azimuth = 0  h = 10
< 175 >                          w = 100  construction = WALL-typ1  ..
< 178 >          window          x =  0
< 179 >                          y = 3.0
< 180 >                          w = 100  h = 3.0
< 181 >                          glass-type = GT-typ1  ..
< 183 >
< 187 >        interior-wall       area = 1275  construction = IWF-typ1
< 188 >                            next-to =  "SP_1_S"  ..
< 190 >
< 192 >        roof  x = 0  y = 0
< 193 >              h = 15
< 194 >              w =  85
< 195 >              construction = ROOF-typ1 ..
<  77 >
<  79 >   "SP_t_E" =
< 170 >
< 171 >     space                x = 100  y = 0  z = 20  azimuth = -90  area = 525
< 172 >                          volume =  5250
< 173 >                          space-conditions = PERIM  ..
< 174 >        exterior-wall     x = 0  y = 0  azimuth = 0  h = 10
< 175 >                          w = 50  construction = WALL-typ1  ..
< 178 >          window          x =  0
< 179 >                          y = 3.0
< 180 >                          w = 50  h = 3.0
< 181 >                          glass-type = GT-typ1  ..
< 183 >
< 187 >         interior-wall    area = 525  construction = IWF-typ1
< 188 >                          next-to =  "SP_1_E"  ..
< 190 >
< 192 >         roof  x = 0  y = 0
< 193 >               h = 15
< 194 >               w =  35
< 195 >               construction = ROOF-typ1 ..
<  84 >                            interior-wall  area = 212.129989624
<  85 >                                           next-to = "SP_t_N"
<  86 >                                           construction = IWF-typ1  ..
<  87 >                            interior-wall  area = 212.129989624
<  88 >                                           next-to = "SP_t_S"
<  89 >                                           construction = IWF-typ1  ..
<  91 >
<  92 >  "SP_t_W" =
< 170 >
< 171 >    space               x = 0  y = 50  z = 20  azimuth = 90  area = 525
< 172 >                        volume =  5250
< 173 >                        space-conditions = PERIM  ..
< 174 >        exterior-wall   x = 0  y = 0  azimuth = 0  h = 10
< 175 >                        w = 50  construction = WALL-typ1  ..
< 178 >          window        x =  0
< 179 >                        y = 3.0
< 180 >                        w = 50  h = 3.0
< 181 >                        glass-type = GT-typ1  ..
< 183 >
< 187 >        interior-wall    area = 525  construction = IWF-typ1
< 188 >                         next-to =  "SP_1_W"  ..
< 190 >
< 192 >        roof  x = 0  y = 0
< 193 >              h = 15
< 194 >              w =  35
< 195 >              construction = ROOF-typ1 ..
<  96 >                           interior-wall  area = 212.129989624
<  97 >                                          next-to = "SP_t_N"
<  98 >                                          construction = IWF-typ1  ..
<  99 >                           interior-wall  area = 212.129989624
< 100 >                                          next-to = "SP_t_S"
< 101 >                                          construction = IWF-typ1  ..
< 103 >
< 104 >  $-------- Core space --------$
< 108 >     "SP_t_C" =  space  x = 15  y = 15
< 109 >                                   z = 20  azimuth = 0  area = 1400
< 110 >                                   volume =  14000
< 111 >                                   space-conditions=CORE  ..
< 112 >          interior-wall  area =  700
< 113 >                         next-to = "SP_t_S"
< 114 >                         construction = IWF-typ1  ..
< 115 >          interior-wall  area =  700
< 116 >                         next-to = "SP_t_N"
< 117 >                            construction = IWF-typ1  ..
< 118 >               interior-wall area =  200
< 119 >                             next-to = "SP_t_E"
< 120 >                             construction = IWF-typ1  ..
< 121 >               interior-wall area =  200
< 122 >                             next-to = "SP_t_W"
< 123 >                             construction = IWF-typ1  ..
< 124 >
< 128 >                 interior-wall       area = 1400
< 129 >                                     next-to =  "SP_1_C"
< 130 >                                     construction = IWF-typ1  ..
< 132 >
< 134 >                 roof          x = 15  y = 15
< 135 >                               h =  20
< 136 >                               w =  70
< 137 >                               construction = ROOF-typ1  ..
<  67 >
<  68 > end ..
<  69 > compute LOADS ..
```