Documentation of ESP-r Geometry File

Edited by Dr. Jon W. Hand
Energy Systems Research Unit
of the University of Strathclyde

Version of 28 May 2010

Introduction

ESP-r has traditionally used a distributed file store to hold the details of ESP-r models. This includes the use of different folders for different type of data e.g. zone composition is held separately from control definitions and description of networks. It also separates data for different zones into separate files.

One of the key model files within this distributed file structure is the zone geometry file. This document covers version 1.1 and is intended to provide technical support for those who archive models or wish to generate this file format via 3rd party software.

Unlike the ESP-r META file format, the zone geometry file is made up of explicit vertices, polygons, surface attributes rather than META components (compact descriptions of entities). Version 1.1 of the zone geometry file follows the following syntax rules:

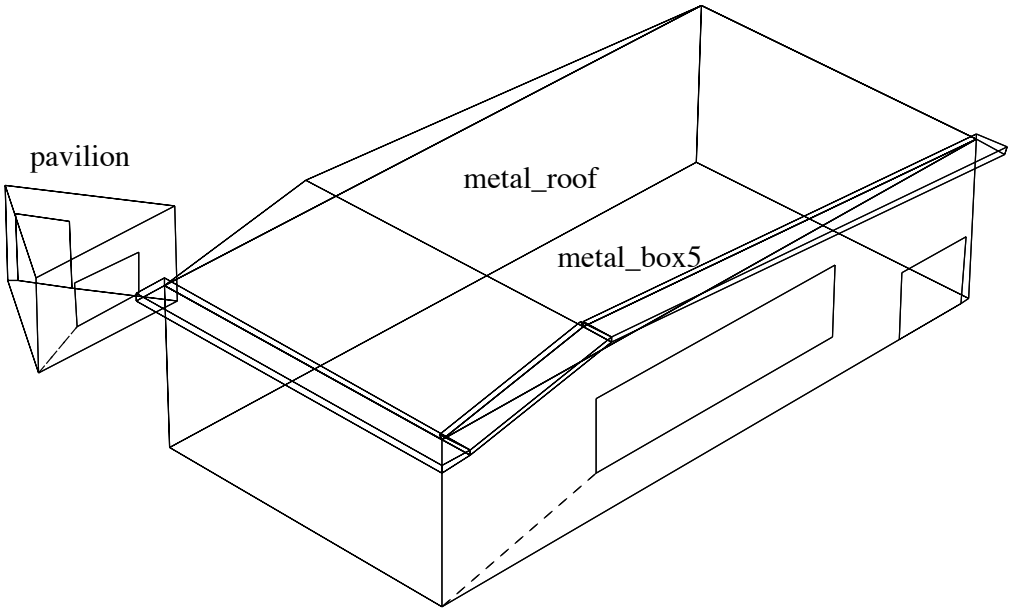
- use a tag-data format for as many entities as possible as well as delineating related groups of data with topic start and end markers
- new concepts are added to both the META file and the standard geometry file in a similar format and with a common data structure

The evolution of ESP-r is incremental and as new concepts become important within the ESP-r community the syntax will evolve. Separate from the syntax of the file is the underlying data structure. This is gradually transitioning from a look-at-one-zone-at-a-time to holding the whole of the model composition in memory.

Overview

This document provides an overview of the geometry file as well as an details of its syntax and the relationships between entities so that others can generate syntactically correct files.

An example file is shown below which creates a rectangular building with a separate sloped roof space and a small pavilion on the north side via standard analog components. The figure below shows the layout of the model to be created.



Project: Creates a three zone model from META descriptions

Figure 1: Overall view of model rooms

There are three general topologies of rooms included in Figure 1. A room named metal_box5 which might have started out as a /fBbox shape/fR into which a window and a door was inserted. A separate triangular room has been created on the north side and it could have started from an /fBextruded shape/fR with a window inserted into one surface and a door inserted into another. The roof space zone could not have been derived from a box or an extrusion and is composed from /fBgeneral polygons/R.

Any surface which is thermally important in the project must be represented via an attributed polygon. There are no simplified thermophysical entities in ESP-r although some third party tools may include doors and windows which are differently treated.

In the case of the metal_box5 room and the triangular room, 3rd party software must convert its descriptions for windows and doors into the vertices and polygons used in the ESP-r zone geometry file.

Details of the box shaped room are shown in Figure 2, the extruded room are shown in Figure 3 and for the poly shape room in Figure 4. These can be compared with the tokens in the META file (see the separate document).

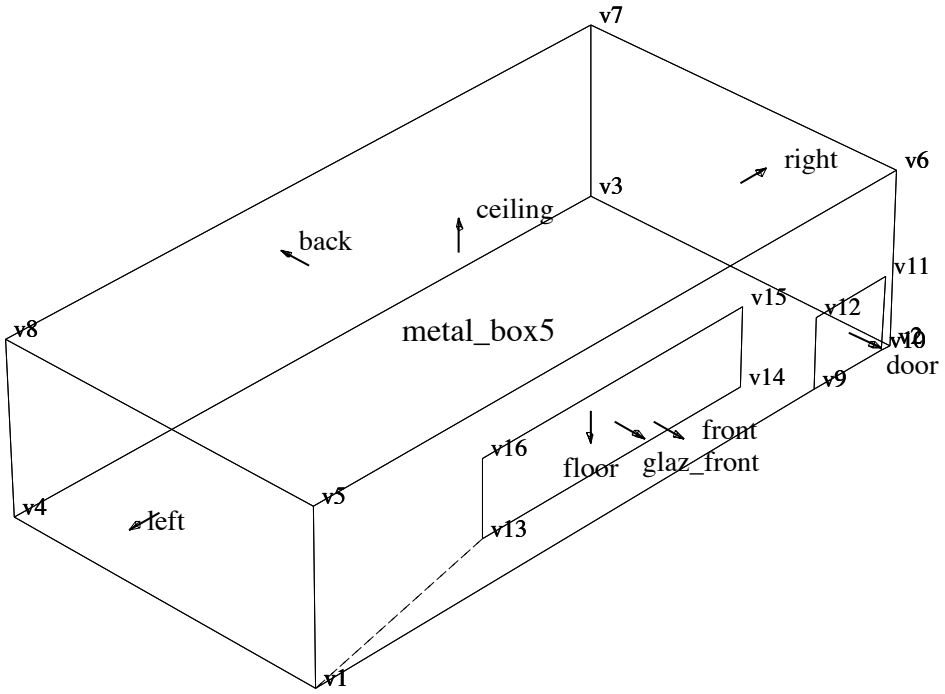


Figure 2: Details of the box shaped room

```
Zone metal_box5 ( 1) is composed of 8 surfaces and 16 vertices.
It encloses a volume of 1000.m^3 of space, with a total surface
area of 700.m^2 & approx floor area of 200.m^2
metal_box5 auto-generated for shape box & attributes.
There is 300.00m2 of exposed surface area, 300.00m2 of which is vertical.
Outside walls are 140.00 % of floor area & avg U of 0.448 & UA of 125.30
Glazing is 10.000 % of floor & 6.6666 % facade with avg U of 2.811 & UA of 56.213

A summary of the surfaces in metal_box5( 1) follows:

Sur| Area  |Azim|Elev| surface     |    geometry       | construction |environment
   | m^2   |deg |deg | name        |optical|locat| use | name         |other side
 1  74.8    180.   0. front        OPAQUE   VERT -     extern_wall  ||< external
 2  50.0     90.   0. right        OPAQUE   VERT -     extern_wall  ||< external
 3  100.      0.   0. back         OPAQUE   VERT -     extern_wall  ||< external
 4  50.0    270.   0. left         OPAQUE   VERT -     extern_wall  ||< external
 5  200.      0.  90. ceiling      OPAQUE   CEIL -     susp_ceil    ||< base:metal_roof
 6  200.      0. -90. floor        OPAQUE   FLOR -     floor_1      ||< ground profile  1
 7  5.25    180.   0. door         OPAQUE   VERT DOOR  door         ||< external
 8  20.0    180.   0. glaz_front   DCF7671_ VERT C-WIN dbl_glz      ||< external
```

The ESP-r Project Manager has one input mode which begins with a box shape and this allows a limited number of parameters to be expanded into the above entities.

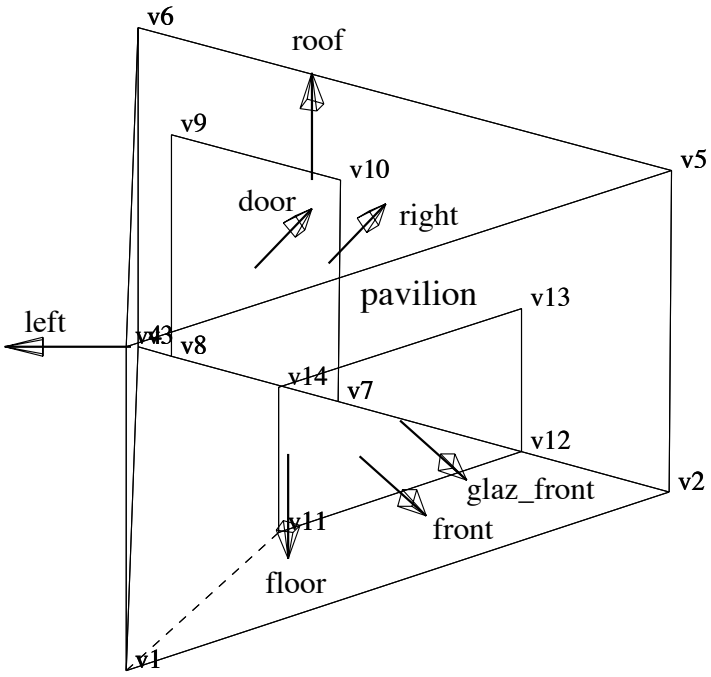


Figure 3: Details of the extrusion derived room

```
Zone pavilion ( 3) is composed of 7 surfaces and 14 vertices.
It encloses a volume of 30.0m^3 of space, with a total surface
area of 63.3m^2 & approx floor area of 10.0m^2
pavilion auto-generated for shape extrude & attributes.
There is 53.302m2 of exposed surface area, 43.302m2 of which is vertical.
Outside walls are 403.02 % of floor area & avg U of 0.621 & UA of 25.035
Flat roof is 100.00 % of floor area & avg U of 1.799 & UA of 17.992
Glazing is 30.000 % of floor & 6.9281 % facade with avg U of 2.811 & UA of 8.4319

A summary of the surfaces in pavilion( 3) follows:

Sur| Area  |Azim|Elev| surface     |    geometry       | construction |environment
   | m^2   |deg |deg | name        |optical|locat| use | name         |other side
 1  12.0    180.   0. front        OPAQUE   VERT -     extern_wall  ||< external
 2  11.0     58.   0. right        OPAQUE   VERT -     extern_wall  ||< external
 3  14.2    302.   0. left         OPAQUE   VERT -     extern_wall  ||< external
 4  10.0      0.  90. roof         OPAQUE   CEIL -     roof_1       ||< external
 5  10.0      0. -90. floor        OPAQUE   FLOR -     floor_1      ||< ground profile  1
 6  3.15     58.   0. door         OPAQUE   VERT DOOR  door         ||< external
 7  3.00    180.  -0. glaz_front   DCF7671_ VERT C-WIN dbl_glz      ||< external
```

The extrude type of room is topologically similar to the existing floor plan extrusion which users create interactively. The conventions for edge ordering (anti-clockwise looking down on the floor plate) are the same as are the limitations on the complexity of the floor plate (no more than 32 edges).

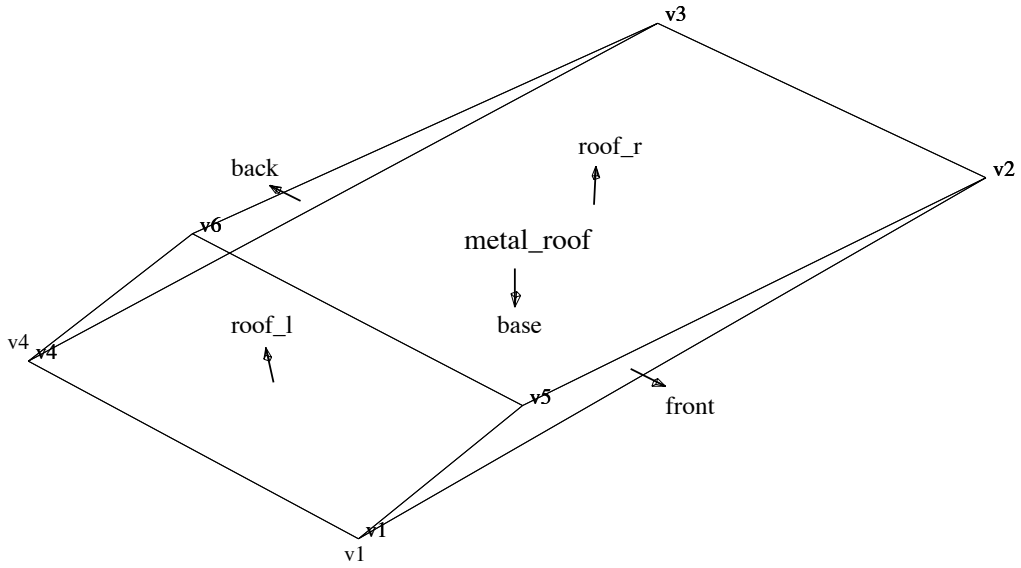


Figure 4: Details of the polygon derived room

```
Zone metal_roof ( 2) is composed of 5 surfaces and 6 vertices.
It encloses a volume of 100.m^3 of space, with a total surface
area of 421.m^2 & approx floor area of 200.m^2
metal_roof auto-generated for shape poly & attributes.
There is 221.32m2 of exposed surface area, 20.000m2 of which is vertical.
Outside walls are 10.000 % of floor area & avg U of 0.393 & UA of 7.8541
Sloped roof is 100.66 % of floor area & avg U of 1.799 & UA of 362.23

A summary of the surfaces in metal_roof( 2) follows:

Sur| Area  |Azim|Elev| surface    |    geometry       | construction |environment
   | m^2   |deg |deg | name       |optical|locat| use | name         |other side
 1  200.      0. -90. base         OPAQUE   FLOR -     susp_ceil    ||< ceiling:metal_box5
 2  10.0     180.   0. front        OPAQUE   VERT -     extern_wall  ||< external
 3  150.      90.  86. roof_r       OPAQUE   SLOP -     roof_1       ||< external
 4  10.0       0.   0. back         OPAQUE   VERT -     extern_wall  ||< external
 5  51.0     270.  79. roof_l       OPAQUE   SLOP -     roof_1       ||< external

Surfaces (all applicable) for shading analysis:
front roof_r back roof_l
No insolation analysis requested.
Block X-coord Y-coord Z-coord DX VAL. DY VAL. DZ VAL. Orien
  1     0.0    -1.0     5.0     5.1     1.0     0.2     0.0    11.3 roof_l_ovhs roof
  2     5.0    -1.0     6.0    15.0     1.0     0.2     0.0    -3.8 roof_r_ovhs roof
  3    -1.0    -1.0     5.0     1.0    11.0     0.2     0.0     0.0 roof_l_w roof
```

The poly shape room is similar in format to the native ESP-r zone geometry file and it is able to represent any box or extrude initial shape as well as enclosures of arbitrary complexity. It is governed by the conventions of the GEN shape zone in ESP-r. If a room cannot be created from an initial box or extrude shape then the poly shape type can be used. In the example the intent is a roof space with a sloped roof because neither of them includes a provision for sloped surfaces.

The input geometry files that generated the above model are listed below. After the file contents each of the tokens will be explained.

```
*Geometry 1.1,GEN,metal_box5 # tag version, format, zone name
*date Wed Oct  7 11:39:23 2009  # latest file modification
metal_box5 auto-generated for shape poly & attributes.
# tag, X co-ord, Y co-ord, Z co-ord
*vertex,0.00000,0.00000,0.00000  #   1
*vertex,20.00000,0.00000,0.00000  #   2
*vertex,20.00000,10.00000,0.00000  #   3
*vertex,0.00000,10.00000,0.00000  #   4
*vertex,0.00000,0.00000,5.00000  #   5
*vertex,20.00000,0.00000,5.00000  #   6
*vertex,20.00000,10.00000,5.00000  #   7
*vertex,0.00000,10.00000,5.00000  #   8
*vertex,17.20000,0.00000,0.00000  #   9
*vertex,19.70000,0.00000,0.00000  #  10
*vertex,19.70000,0.00000,2.10000  #  11
*vertex,17.20000,0.00000,2.10000  #  12
*vertex,5.52790,0.00000,1.38200  #  13
*vertex,14.47210,0.00000,1.38200  #  14
*vertex,14.47210,0.00000,3.61800  #  15
*vertex,5.52790,0.00000,3.61800  #  16
#
# tag, number of vertices followed by list of associated vert
*edges,14,1,9,12,11,10,2,6,5,1,13,16,15,14,13  #  1
*edges,4,2,3,7,6  #  2
*edges,4,3,4,8,7  #  3
*edges,4,4,1,5,8  #  4
*edges,4,5,6,7,8  #  5
*edges,6,1,4,3,2,10,9  #  6
*edges,4,9,10,11,12  #  7
*edges,4,13,14,15,16  #  8
#
# surf attributes:
#  surf name, surf position VERT/CEIL/FLOR/SLOP/UNKN
#  child of (surface name), useage (pair of tags)
#  construction name, optical name
#  boundary condition tag followed by two data items
*surf,front,VERT,-,-,-,extern_wall,OPAQUE,EXTERIOR,00,00  #   1 ||< external
*surf,right,VERT,-,-,-,extern_wall,OPAQUE,EXTERIOR,00,00  #   2 ||< external
*surf,back,VERT,-,-,-,extern_wall,OPAQUE,EXTERIOR,00,00  #   3 ||< external
*surf,left,VERT,-,-,-,extern_wall,OPAQUE,EXTERIOR,00,00  #   4 ||< external
*surf,ceiling,CEIL,-,-,-,susp_ceil,OPAQUE,ANOTHER,02,01  #   5 ||< not_known:not_known
*surf,floor,FLOR,-,-,-,floor_1,OPAQUE,GROUND,01,00  #   6 ||< ground profile  1
*surf,door,VERT,-,-,-,door,OPAQUE,EXTERIOR,00,00  #   7 ||< external
*surf,glaz_front,VERT,-,-,-,dbl_glz,DCF7671_06nb,EXTERIOR,00,00  #   8 ||< external
#
*insol,3,0,0,0  # default insolation distribution
#
# shading directives
*shad_calc,none  # no temporal shading requested
#
*insol_calc,none  # no insolation requested
#
*base_list,1,6,   200.00 0  # zone base list

---------------

*Geometry 1.1,GEN,pavilion # tag version, format, zone name
*date Wed Oct  7 11:39:23 2009  # latest file modification
pavilion auto-generated for shape poly & attributes.
# tag, X co-ord, Y co-ord, Z co-ord
*vertex,0.00000,15.00000,0.00000  #   1
*vertex,5.00000,15.00000,0.00000  #   2
*vertex,2.50000,19.00000,0.00000  #   3
*vertex,0.00000,15.00000,3.00000  #   4
*vertex,5.00000,15.00000,3.00000  #   5
*vertex,2.50000,19.00000,3.00000  #   6
*vertex,3.45400,17.47360,0.00000  #   7
*vertex,2.65900,18.74560,0.00000  #   8
*vertex,2.65900,18.74560,2.10000  #   9
*vertex,3.45400,17.47360,2.10000  #  10
*vertex,1.38200,15.00000,0.82920  #  11
*vertex,3.61800,15.00000,0.82920  #  12
*vertex,3.61800,15.00000,2.17080  #  13
*vertex,1.38200,15.00000,2.17080  #  14
```

```
#
# tag, number of vertices followed by list of associated vert
*edges,10,1,2,5,4,1,11,14,13,12,11  #  1
*edges,8,2,7,10,9,8,3,6,5  #  2
*edges,4,3,1,4,6  #  3
*edges,3,4,5,6  #  4
*edges,5,1,3,8,7,2  #  5
*edges,4,7,8,9,10  #  6
*edges,4,11,12,13,14  #  7
#
# surf attributes:
#  surf name, surf position VERT/CEIL/FLOR/SLOP/UNKN
#  child of (surface name), useage (pair of tags)
#  construction name, optical name
#  boundary condition tag followed by two data items
*surf,front,VERT,-,-,-,extern_wall,OPAQUE,EXTERIOR,00,00  #   1 ||< external
*surf,right,VERT,-,-,-,extern_wall,OPAQUE,EXTERIOR,00,00  #   2 ||< external
*surf,left,VERT,-,-,-,extern_wall,OPAQUE,EXTERIOR,00,00  #   3 ||< external
*surf,roof,CEIL,-,-,-,roof_1,OPAQUE,EXTERIOR,00,00  #   4 ||< external
*surf,floor,FLOR,-,-,-,floor_1,OPAQUE,GROUND,01,00  #   5 ||< ground profile  1
*surf,door,VERT,-,-,-,door,OPAQUE,EXTERIOR,00,00  #   6 ||< external
*surf,glaz_front,VERT,-,-,-,dbl_glz,DCF7671_06nb,EXTERIOR,00,00  #   7 ||< external
#
*insol,3,0,0,0  # default insolation distribution
#
# shading directives
*shad_calc,none  # no temporal shading requested
#
*insol_calc,none  # no insolation requested
#
*base_list,1,5,    10.00 0  # zone base list

----------------

*Geometry 1.1,GEN,metal_roof # tag version, format, zone name
*date Wed Oct  7 11:55:51 2009  # latest file modification
metal_roof auto-generated for shape poly & attributes.
# tag, X co-ord, Y co-ord, Z co-ord
*vertex,0.00000,0.00000,5.00000  #   1
*vertex,20.00000,0.00000,5.00000  #   2
*vertex,20.00000,10.00000,5.00000  #   3
*vertex,0.00000,10.00000,5.00000  #   4
*vertex,5.00000,0.00000,6.00000  #   5
*vertex,5.00000,10.00000,6.00000  #   6
#
# tag, number of vertices followed by list of associated vert
*edges,4,1,4,3,2  #  1
*edges,3,1,2,5  #  2
*edges,4,2,3,6,5  #  3
*edges,3,3,4,6  #  4
*edges,4,4,1,5,6  #  5
#
# surf attributes:
#  surf name, surf position VERT/CEIL/FLOR/SLOP/UNKN
#  child of (surface name), useage (pair of tags)
#  construction name, optical name
#  boundary condition tag followed by two data items
*surf,base,FLOR,-,-,-,susp_ceil,OPAQUE,ANOTHER,01,05  #   1 ||< ceiling:metal_box5
*surf,front,VERT,-,-,-,extern_wall,OPAQUE,EXTERIOR,0,0  #   2 ||< external
*surf,roof_r,SLOP,-,-,-,roof_1,OPAQUE,EXTERIOR,0,0  #   3 ||< external
*surf,back,VERT,-,-,-,extern_wall,OPAQUE,EXTERIOR,0,0  #   4 ||< external
*surf,roof_l,SLOP,-,-,-,roof_1,OPAQUE,EXTERIOR,0,0  #   5 ||< external
#
*insol,3,0,0,0  # default insolation distribution
#
# shading directives
*shad_calc,all_applicable   4 # list of surfs
  2  3  4  5
#
*insol_calc,none  # no insolation requested
#
*base_list,1,1,   200.00 0  # zone base list
#
# block entities:
#  *obs = obstructions
*block_start,20 20 # geometric blocks
*obs3,0.000,-1.000,5.000,5.099,1.000,0.200,0.000,11.310,0.000,roof_l_ovhs,roof  # block  1
*obs3,5.000,-1.000,6.000,15.033,1.000,0.200,0.000,-3.814,0.000,roof_r_ovhs,roof  # block  2
```

```
*obs,-1.000,-1.000,5.000,1.000,11.000,0.200,0.000,roof_l_w,roof  # block  3
*end_block

--------------
```

The tokens in the ESP-r geometry file are listed below. Where there are start and end marks these are discussed together. Tokens are almost always followed by a comma and a comma is typically used to separate data. Unless specified, the order of the tokens is not critical (3rd party developers warning - testing alternative layouts is recommended).

```
*Geometry 1.1,GEN,metl_roof
. . .
```

The *Geometry line signals the file type as well as holding high level information.

- *Geometry - initial tag for file type
- 1.1 - version of the file
- GEN - tag to signal general polygons will follow
- text - user label for the zone (up to 12 characters as a single word with no spaces or punctuation).

```
*date Wed Oct  7 11:55:51 2009  # latest file modification
metal_roof auto-generated for shape poly & attributes.
```

- *date - tag for the date of file creation
- text - a combination of strings and numbers making up the date

The *date line is followed by a single line of documentation about the zone (up to 64 characters are retained).

Geometric related data

The **GEN** shape is the most general form of zone description and is composed of a number of coordinates in space followed by surface definitions which are defined as a list of edges which define arbitrary polygons. These can be used to explicitly represent the full geometric complexity supported by ESP-r. If you have a sloped ceiling or a window which is not rectangular or wish to include complex surfaces floating within the zone to represent thermal mass this can be accommodated.

```
*rotate,10.00,0.00,0.00
```

or

```
*previous_rotate,10.00,0.00,0.00
```

A *rotate or *previous_rotate line can happen anywhere within the geometry file. Each has three real data which are the degrees (positive is anti-clockwise) of rotation to be applied to the zone followed by the X and Y coordinate to rotate around. A *previous_rotate token is optional.

A *rotate is a request to apply the rotation as the geometry file is scanned (after which this line would become a *previous_rotate to signal that the change has been applied. Third party applications can use the *rotate tag to apply transforms to an initial set of coordinates.

*previous_rotate tag records the users prior request for rotation (for a possible un-do command). Both *rotate and *previous_rotate are optional.

```
# tag, X co-ord, Y co-ord, Z co-ord
*vertex,0.00000,0.00000,5.00000  #   1
```

```
*vertex,20.00000,0.00000,5.00000  #   2
*vertex,20.00000,10.00000,5.00000  #   3
*vertex,0.00000,10.00000,5.00000  #   4
*vertex,5.00000,0.00000,6.00000  #   5
*vertex,5.00000,10.00000,6.00000  #   6
. . .
```

For a GEN shape type one or more *vertex lines are expected. There are three data tokens which represent the X, Y and Z co-ordinates in metres. The minimum number of *vertex lines is one although three is a more typical minimum number (e.g to define a single triangular polygon.

There should be one *vertex line for each vertex used in the zone. All vertices should be unique points in space. The order in which the vertices are defined is important. The first *vertex line is assumed to represent the 1st vertex and the 2nd *vertex line is the 2nd vertex of the zone. Polygons are made up of edges which are defined by a list of indices of vertices. If one of the items in a list is a '2' this is interpreted as the second *edges definition.

If two *vertex lines are identical or differ by a fraction of a mm then it is likely that the topology rules of ESP-r will fail.

```
# tag, number of vertices followed by list of associated vert
*edges,4,1,4,3,2  #  1
*edges,3,1,2,5    #  2
*edges,4,2,3,6,5  #  3
*edges,3,3,4,6    #  4
. . .
```

The *edges line starts with an integer representing the number of vertices which are used to make up the edges of the polygon. This is followed by a list of vertices which make up the edges of the polygon. For example, in the example above '4,1,4,3,2' says there are 4 vertices, the first point uses the first vertex data defined for this zone, the second point uses the fourth vertex data, the third point uses the third vertex and the fourth point uses the second vertex. It is assumed that the last edge of the polygon is from the last vertex coordinate to the coordinate of the first vertex.

The order of the vertices is important. It is used to define the direction of the surface normal using the *right hand rule*. If viewed from the outside the edges should be defined anti-clockwise.

```
#
# surf attributes:
#  surf name, surf position VERT/CEIL/FLOR/SLOP/UNKN
#  child of (surface name), useage (pair of tags)
#  construction name, optical name
#  boundary condition tag followed by two data items
*surf,base,FLOR,-,-,-,susp_ceil,OPAQUE,ANOTHER,01,05  #   1 ||< ceiling:metal_box5
*surf,front,VERT,-,-,-,extern_wall,OPAQUE,EXTERIOR,0,0  #   2 ||< external
*surf,roof_r,SLOP,-,-,-,roof_1,OPAQUE,EXTERIOR,0,0  #   3 ||< external
*surf,back,VERT,-,-,-,extern_wall,OPAQUE,EXTERIOR,0,0  #   4 ||< external
. . .
```

Each surface line has a number of attributes:

- ***surf** - tag for surface data
- text - the name of the surface (up to 12 char with no spaces)
- **VERT CEIL FLOR SLOP UNKN** - key words indicating the approximate orientation of the surface. VERT is within a few degrees of vertical, CEIL the outside face is facing up (or within a few degrees), FLOR the outside is facing down (or within a few degrees), SLOP is for sloped orientations and UNKN is for cases where no preference or analysis has been done to classify the surface.

- text the name of a surface which is the parent of this surface or a '-' to signal that the surface has no parent (another surface that wraps around at least 3 sides.

The 5th and 6th tokens on the line define the **use** of the surface. Currently this is only applicable to UK code compliance tasks (so that there is an automatic process for generating 'notional' versions of a model that strictly comply with building code requirements. It should be possible to use these attributes to support compliance tasks for other regions. Otherwise, surface use attributes are for **information purposes**.

A '-,-' signals an ordinary surface with no special attributes. The 2nd token can indicate the type of air flow associated with the surface. A future version of ESP-r will use these tags to help in the automation of air flow networks.

Other combinations are listed below:

- **-,-** an ordinary surface with no specified usage attributes or for which no usage has been defined. This is often the initial state until the user has expressed an opinion.
- **WALL,-** if a near vertical surface this would be composed of a code compliant construction. If within 20 degrees of the surface would be composed of a code compliant roofing material. The 2nd token is not used and is always a '-'.
- **DOOR,-** normal door (inside or outside) which uses code compliant constructions
- **DOOR,CLOSED** as above but signaling it is closed for purposes of air movement
- **DOOR,UNDERCUT** as above but with a crack under the door for air movement
- **DOOR,OPEN** as above but open for purposes of air movement
- **DOOR,BIDIR** as above but with bi-directional air flows assumed
- **P-DOOR,-** a fire door for personnel egress with specific code compliant attributes. The 2nd token can also be CLOSED, UNDERCUT, OPEN, BIDIR
- **H-DOOR,-** a high usage (entrance) door. The 2nd token can also be CLOSED, UNDERCUT, OPEN, BIDIR
- **V-DOOR,-** a door for vehicle access. The 2nd token can also be CLOSED, UNDERCUT, OPEN, BIDIR
- **FRAME,-** a frame associated with an inside door or glazing. The 2nd token can also be CLOSED, CRACK, OPEN.
- **F-FRAME,-** marks a code compliant frame within the building facade. The 2nd token can also be CLOSED, CRACK, OPEN.
- **C-WINDOW,-** marks a code compliant window in terms of its construction or its size. The 2nd token can also be CLOSED, CRACK, OPEN, SASH, BIDIR to identify air flow characteristics.
- **D-WINDOW,-** marks a display window (in a commercial building) which does not need to confirm to specific size or construction requirements.
- **S-WINDOW,-** marks a code compliant skylight in terms of its construction or its size. The 2nd token can be CLOSED, CRACK, OPEN, SASH, BIDIR.
- **WINDOW,-** any other window on the inside or the facade. The 2nd token can be CLOSED, CRACK, OPEN, SASH, BIDIR.
- **GRILL,SOURCE** a surface associated with an inlet grill
- **GRILL,EXTRACT** a surface associated with an extract grill
- **SMOKEVENT,-** a surface associated with a smoke ventilation opening. The 2nd token is always '-'.
- **ROOF,FLAT** a roof surface with a 0-20 degree incline.
- **ROOF,PITCHED** a roof surface with greater than 20 degree incline.
- **FICT,-** a surface which is-not-there in terms of its composition (low mass, high transmission of light etc.). The 2nd token cal also be CLOSED, CRACK, OPEN, BIDIREC.
- **FLOOR,-** a floor surface inside the building
- **FLOOR,EXTGRND** a floor surface connected to a ground boundary type
- **FURNI,-** a surface representing internal mass. The 2nd token is always '-'.
- **PARTN,-** a surface which is a partition (wall, floor, ceiling) between two thermal zones. The 2nd token is always '-'.

- **STRUC,-** a surface associated with the building structure (beam or column). The 2nd token is always '-'.

The usage tokens are followed by construction and optical information as well as the associated boundary condition.

- text - name of the construction (up to 12 char with no spaces). The name of the construction MUST match a name in the construction database associated with the model. The underlying data structure accommodates construction names up to 32 characters and with spaces. Currently only the initial 12 characters are used.
- text - either 'OPAQUE' or 'CFC' for a complex fenestration or 'TRAN' for an unspecified optical property (will use the one which is associated with the constructions) or the name of the optical property (up to 12 char with no spaces in the optical database) which is associated with the surface construction.

The last three data in the *surf line are a key word and two integers that define the boundary conditions at the other face of the surface. The values use the same order and syntax as are found in the model configuration connections file.

- **EXTERIOR,00,00** a surface facing the outside. The 2nd and 3rd tokens are always 00.
- **ANOTHER,02,01** a surface facing another zone. The 2nd token is the index of the other zone (e.g. a 3 is the 3rd zone defined in the model) and the 3rd token is the index of the surface in the other zone (e.g. a 4 is the 4th surface defined in the other zone).
- **ADIABATIC,00,00** is a boundary where no flux passes. This is often used when a portion of the building is defined and the adjacent portion is not considered. The 2nd and 3rd tokens are always 00.
- **SIMILAR,00,00** is a boundary where the other side is at the same temperature as the zone we are in (at any moment in time). The 2nd token is an offset in degrees C and the 3rd token is an offset in Watts. For example if the boundary is always one degree warmer than the current zone then a non-zero value would be entered in the 2nd token.
- **UNKNOWN,00,00** no boundary condition has been set for this surface. A simulation cannot be run until a boundary condition is set.
- **BASESIMP,00,00** the surface boundary is part of a BASESIMP basement definition. The 2nd and 3rd tokens are used to clarify which part of the BASESIMP definition is associated with the current surface. There are several dozen combinations which are defined via the 2nd and 3rd tokens (and these are not yet included in this document).
- **GROUND,01,00** the surface is in contact with the ground. The 2nd and 3rd tokens defines which type of ground and which ground temperature profile to use (see the model connections file for specifics).
- **CONSTANT,00,00** the surface faces a constant temperature defined by the 2nd token and a constant radiation (Watts) defined by the 3rd token.
- **IDENT_CEN,00,00** the surface has a CEN standard identical boundary condition. This is for use in CEN validation studies and should not be used for other purposes. The two numbers which follow the IDENT_CEN are always zero.

Essentially the agent creating the zone geometry file file is assumed to know the 'topology' of the model and to follow the descriptive rules which apply to an ESP-r model *.cnn file. One way to check the logic of a third party application as it generates a geometry file is to create an equivalent model in ESP-r and look for differences.

Directives for increasing model resolution

ESP-r models include facilities to increase the resolution of assessments for the distribution of solar radiation.

```
# shading directives
*shad_calc,none  # no temporal shading requested
# insolation directives
*insol_calc,none  # no insolation requested
```

ESP-r supports the calculation of shading patterns on exterior surfaces at each hour of the day for a typical day in each month of the simulation as well as the distribution of direct solar radiation within zones (insolation). Shading calculations can be undertaken if there are shading obstructions defined in the model (see the next section).

Whether or not shading patterns have been calculated the user can request that the pattern of solar insolation is calculated. There are thus separate tokens in the META file which provide directives as to how shading and

insolation should be treated. In the above example the directive token **none** stipulates that the calculation is not requested and that the default assumptions for shading and/or insolation is to be used within the simulation.

```
# shading directives
*shad_calc,all_applicable   4 # list of surfs
 2  3  4  5
# insolation directives
*insol_calc,all_applicable  1  # list of surfaces
 7
```

In the above example the token is **all_applicable** and this is followed by an integer (the number of surfaces which are to be considered for shading). The next line contains the list of surfaces for which shading should be calculated (e.g. 2 is the second surface in the zone, 3 is the third surface in the zone). Surfaces in this list must have an**outside** boundary condition.

In the above *insol_calc line the **all_applicable** has a similar syntax. In this case it identifies **transparent surfaces** which have an outside boundary condition and the following data line is a list of transparent surfaces (there happens to be only one surface in the example above).

Block objects

ESP-r includes several geometric objects which are thermophysically inactive (they are not part of the convective exchanges within a room) but which are used to represent visual entities or entities which block direct or diffuse solar radiation.

```
*block_start,20,20
*obs,-1.00,-1.00,5.00,1.00,11.00,0.20,0.00,roof_l_w,roof  # block  3
. . .
*end_block
```

- ***block_start** - tag at start of the section
- number - X shading grid density (number of grid points)
- number - Z shading grid density (number of grid points)
- ***end_block** tag at the end of the block object section

Solar obstructions are defined via one or more *obs lines within a *block_start and *end_block section. For a simple obstruction There are 9 data tokens as follows:

- ***obs** - the tag
- number - X origin of the block (m)
- number - Y origin of the block (m)
- number - Z origin of the block (m)
- number - length (m) along east axis if no rotation
- number - depth (m) along north axis if no rotation
- number - height (m) along Z axis
- number - degrees of rotation (positive is anti-clockwise) about the origin.
- text - name of the block (up to 12 char)
- text - name of the construction of the block (up to 12 char) and must point to an existing construction within the construction database used by the model.

In the example above there is a block which is 1m long and 11m wide and 20cm tall with an origin at -1.0 -1.0 5.0 which is not rotated (its length runs along the X axis).

```
*obs3,0.00,-1.00,5.00,5.099,1.00,0.20,0.00,11.310,0.00,roof_l_ovhs,roof  # block  1
```

Solar obstructions which can be rotated in multiple axis are defined via one or more *obs3 lines within the *block_start and *end_block section. There are 11 data tokens. The first 6 are identical to the *obs type. The remaining tokens are as follows:

- ***obs3** - the tag
- number - degrees of rotation (positive is anti-clockwise) about the origin.
- number - degrees of elevation (positive is upwards) between the ground plane and the length of the block.
- number - degrees of tilt (positive is upwards) between the ground plane and the width of the block. This rotation is currently NOT implemented.
- text - name of the block (up to 12 char)
- name of the construction of the block (up to 12 char) and must point to an existing construction within the construction database used by the model.

In the example above, there is a block which is 5.099m long, 1.0m wide and 0.2m high which has an origin at XYZ 0.0 -1.0 5.0 and which is tilted upwards along its length by 11.31 degrees.

Thermal Bridges

ESP-r supports the concept of linear thermal bridges within a model. A linear thermal bridge is a correction factor (sometimes called a psi value) between the performance of a 1D conduction model and a 2D conduction representation from 3rd party software such as THERM. Multiple thermal bridge types can be defined within a *bridge_start and a *end_bridge marker.

```
  *bridge_start,0.000,0.000,9.350,59.827  # user fraction,losspercent W/K, total heat l
  *ukt_bridge,jamb at window or door,  6.400, -0.2100  # type length psi value
  . . .
*end_bridge
```

- ***bridge_start** - tag at the start of thermal bridge section
- number - user fraction thermal bridge (ignore if zero)
- number - user Watts/degree K percent (ignore if zero)
- number - total (derived) heat loss W/K of the thermal bridges
- number - total (derived) UA W/K of the thermal bridges

Each type of thermal bridge has a tag and a description as well as two numerical attributes.

- *ukt_bridge - tag for a linear thermal bridge defined as a psi value (W/m/K) and a length (m). In the future other thermal bridges with different attributes with a different key word could be added.
- text description of the type of thermal bridge. Currently the ESP-r project manager generates several standard blocks of text such as 'jamb at window or door', 'wall-wall (convex corner)' ,'wall-wall (concave corner)'.
- number - total length of this kind of thermal bridge (m)
- number - psi value for this kind of thermal bridge W/m/K

A psi value for a thermal bridge is typically generated in via a detailed 2D conduction analysis in a 3rd party application such as THERM (from LBL) and is beyond the scope of this document.

Surfaces associated with the zone base

```
  *base_list,3,14,21,22,13.63

or
  *base_list,0,4.00,1
```

A zone may have one or more surfaces which make up its base. The definition of which surfaces are associated with the base area of the zone (up to 6) or that the user has supplied a base area are determined by the contents of the *base_list line.

- ***base_list** the tag for this topic
- number - the number of surfaces (up to 6). If zero then the following number will be the user defined area.

- list of numbers the index of each surface in the list
- number - the area (m2) (user supplied or calculated)
- number - zero if a list of surfaces has been given and a one if the user supplied an area value.

Revisions

The following is a summary of changes made to the geometry file format. This section is work-in-progress.