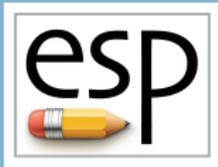


# Engineering Sketch Pad (ESP)



## Training Session 8 Selection & Attribution

**John F. Dannenhoffer, III**

[jfdannen@syr.edu](mailto:jfdannen@syr.edu)  
Syracuse University

**Bob Haines**

[haines@mit.edu](mailto:haines@mit.edu)  
Massachusetts Institute of Technology  
updated for v1.22

- Purpose and Types of Attributes
- Setting Attributes
- Viewing Attributes: DisplayFilter
- Selecting Entities
- Attributes That are Automatically Set
- Csystem
- Editing Attributes: UDPRIM editAttr
- Homework Exercise

- Attributes are meta-data that can be used to tag any entity
- Attributes can be applied to:
  - Bods
  - Faces
  - Edges
  - Nodes
- Attributes can be:
  - one or more integers (reserved for internal use)
  - one or more floating-point numbers
  - a character string

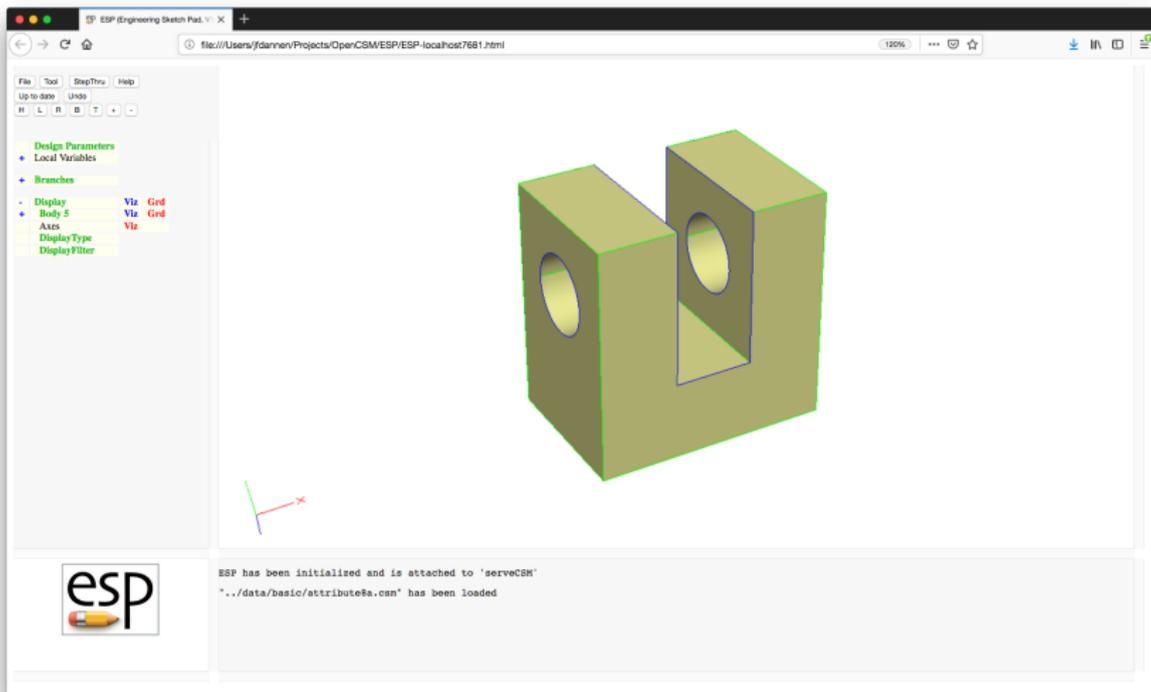
- Attributes can be associated with any Branch that produces a Body or the **SELECT** statement
- Attributes are defined by an **ATTRIBUTE** statement
- Take care when starting an Attribute name with a period (which is reserved for **EGADS**) or an underscore (which is reserved for **OpenCSM**)
- If the first character of the value is a dollar-sign, then the Attribute will contain a character string
- Otherwise the Attribute will contain one or more real (double) values
  - if the value is the name of a multi-valued Parameter, then the Attribute will be multi-valued
  - if the value is a semi-colon-separated list of expressions, then the Attribute will be multi-valued
  - otherwise the Attribute will be a single real (double)

- Global Attributes are set with an **ATTRIBUTE** statement before the first Body is created
- Attributes can be set for a Body (and all newly-created Faces) with an **ATTRIBUTE** statement following the Branch that created the Body
- Attribute can be set on any entity(s) by putting an **ATTRIBUTE** statement following a **SELECT** statement
- Best practice is to set the Attributes as soon as the Body is created (for example via a primitive or grown Body command)



# Attribute Example (1)

Whole configuration





# Attribute Example (2)

.csm file

```
ATTRIBUTE density 2710 # global attribute
```

```
BOX          0 0 0 3 3 2
```

```
ATTRIBUTE tag $block
```

```
BOX          1 1 0 1 2 2
```

```
ATTRIBUTE tag $slot
```

```
SUBTRACT
```

```
CYLINDER -1 2 1 4 2 1 1/2
```

```
ATTRIBUTE tag $hole
```

```
SUBTRACT
```

```
END
```



# Attribute Example (3)

DisplayFilter to tag block

The screenshot shows the ESP (Engineering Sketch Pad) software interface. The main window displays a 3D model of a rectangular block with a circular hole on the left side and a rectangular cutout on the right side. The model is rendered in a light green color. A coordinate system with red, green, and blue axes is visible in the bottom left corner of the 3D view.

The left sidebar contains a tree view with the following items:

- Design Parameters
- Local Variables
- Branches
- Display (Viz) (red)
- Body 5 (Viz) (red)
- Axes (Viz) (red)
- Display Type
- Display Filter

The bottom status bar displays the ESP logo and the following text:

```
ESP has been initialized and is attached to 'serveCRM'  
"../data/basic/attribute$.csm" has been loaded  
Display filtered to "tag" "block"
```



# Attribute Example (4)

DisplayFilter to tag slot

The screenshot shows the ESP (Engineering Sketch Pad) interface. The main window displays a 3D model of a rectangular block with a slot cut through its center. The slot is highlighted in a dark green color, indicating that a display filter has been applied to it. The left sidebar shows the design tree with the following items:

- Design Parameters
- Local Variables
- Branches
- Display (Viz) (red)
- Body 5 (Viz) (red)
- Axes (Viz) (red)
- Display Type
- Display Filter

The bottom status bar contains the ESP logo and the following text:

```
ESP has been initialized and is attached to 'serveGRH'  
"../data/basic/attribute$.cam" has been loaded  
Display filtered to "tag" "block"  
Display filtered to "tag" "slot"
```



# Attribute Example (5)

## DisplayFilter to tag hole

The screenshot shows the ESP (Engineering Sketch Pad) software interface. The main window displays a 3D model of two rectangular blocks with cylindrical holes. The left block is solid grey, and the right block is semi-transparent yellow. The software interface includes a menu bar (File, Tool, StepThru, Help), a toolbar (Up to date, Undo, H, L, R, B, T, +, -), and a left-hand tree view showing the design structure:

- Design Parameters
- Local Variables
- Branches
- Display (Viz, Col)
- Body 5 (Viz, Col)
- Axes (Viz)
- Display Type
- Display Filter

The status bar at the bottom contains the ESP logo and a console window with the following text:

```
ESP has been initialized and is attached to 'serveCRM'  
"../data/basic/attribute$.cam" has been loaded  
Display filtered to "tag" "block"  
Display filtered to "tag" "slot"  
Display filtered to "tag" "hole"
```



# Attribute Example (6)

DisplayFilter to density \*

The screenshot shows the ESP (Engineering Sketch Pad) interface. The main window displays a 3D model of a mechanical part, which is a rectangular block with a central slot and two circular holes on the front face. The model is rendered in a light green color. The left sidebar shows a tree view of the design structure, including Design Parameters, Local Variables, Branches, and Display. The Display section is expanded, showing a list of display filters: Display, Body 5, Axes, Display Type, and Display Filter. The Display Filter is selected, and its properties are shown in the bottom right corner of the interface. The properties include a file path, a list of tags, and a density value.

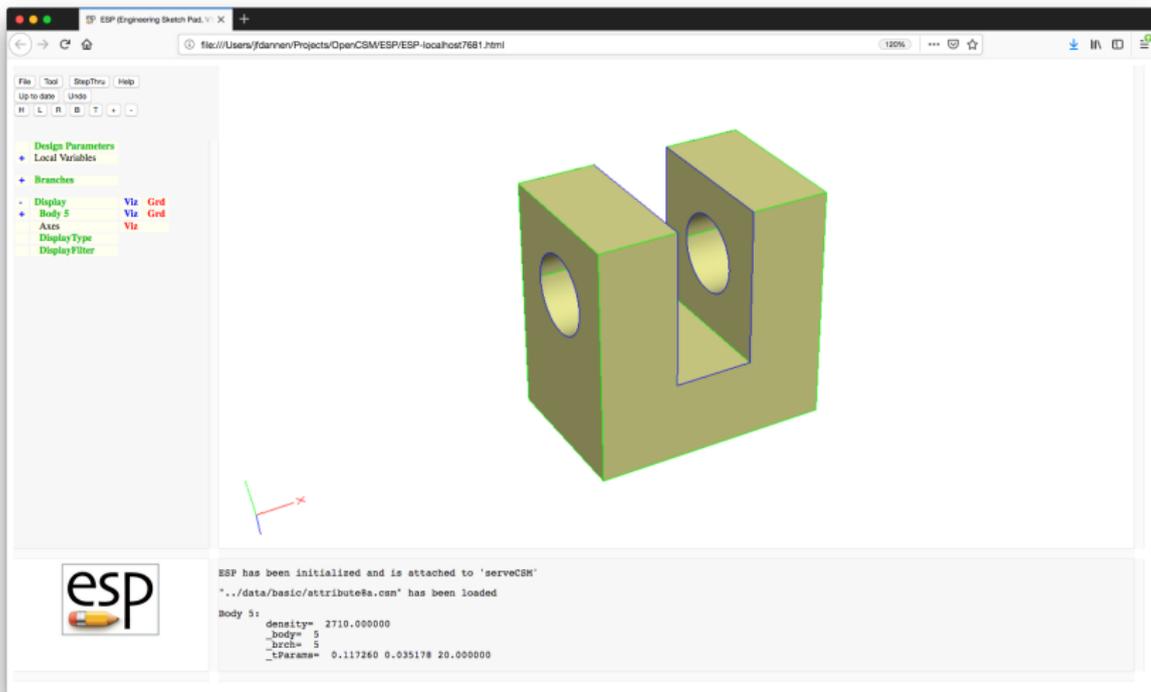
```
../data/basic/attribute8a.csm has been loaded  
Display filtered to "tag" "block"  
Display filtered to "tag" "slot"  
Display filtered to "tag" "hole"  
Display filtered to "density" "*"
```

- Attributes can be viewed in **ESP** in three ways:
  - pressing the mouse in the Tree Window when cursor is over the Body name
  - pressing the  $\wedge$  or **6** key when pointing to a Face, Edge, or Node in the Graphics Window
  - using the **Display Filter** option (at the bottom of the Tree Window)



# Attribute Example (6)

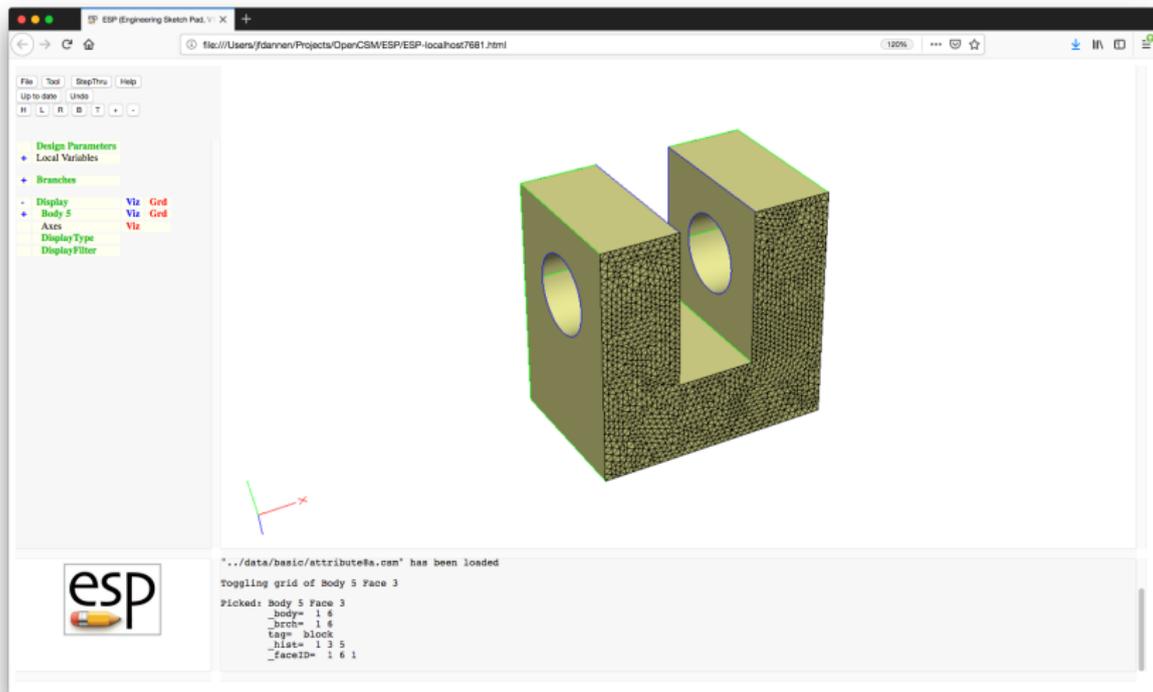
After pressing **Body 5** in TreeWindow

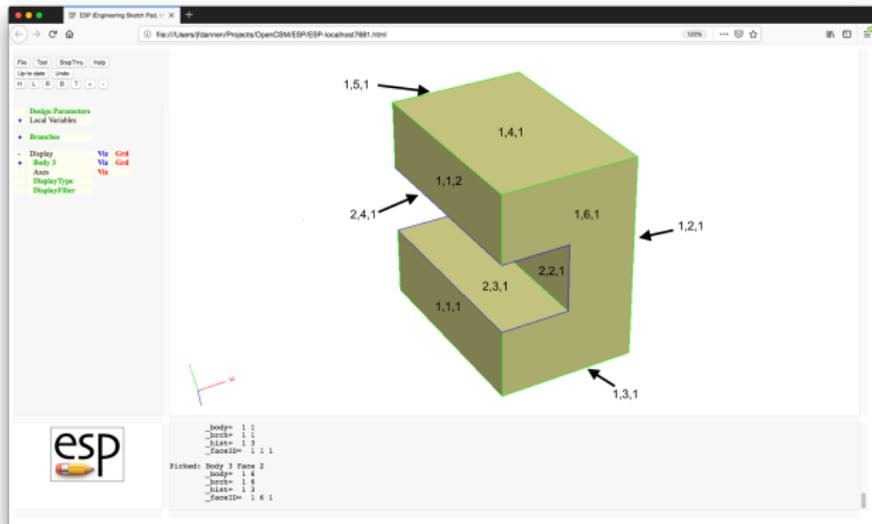




# Attribute Example (6)

After pressing  $\wedge$  on Face with Grid





```

BOX    0  0  0  2  3  3
BOX   -1  1 -1  2  1  5
SUBTRACT
    
```

- FaceID is generated by the Body in which the Face first exists
- EdgeID is generated based upon the ibody/iford of its two adjoining Faces
  - Edge on bottom of front Face has EdgeID 1, 3, 1, 6, 1
  - Edge at bottom of slot & front Face has EdgeID 1, 6, 2, 3, 1

- The **SELECT** statement stores its values in:
  - **@seltype**
    - -1 if only a Body is selected
    - 0 if one or more Nodes are selected
    - 1 if one or more Edges are selected
    - 2 if one or more Faces are selected
  - **@selbody** contains the number of the Body selected
  - **@sellist** contains the list of the Nodes, or Edges, or Faces selected within **@selbody**

- `SELECT BODY` — selects last Body created
- `SELECT BODY ibody` — selects Body `ibody`
- `SELECT BODY -n` — selects the  $n^{\text{th}}$  Body from the top of the Stack
- `SELECT BODY $attrName1 attrValue1 ...` — selects the last Body that matches all the given Attributes

- `SELECT FACE` — selects all Faces in selected Body
- `SELECT FACE iface` — selects Face `iface` in selected Body
  - using this is considered a bad practice since Face numbering may change depending on the version of `OpenCASCADE` that is being used
- `SELECT FACE ibody1 iford1 iseq=1` — selects the Face that has the indicated `ibody1/iford1`
  - as each Face is created, it is marked with the Body in which it was created and the face-order in that Body. This is the preferred technique.
- ...

- ...
- `SELECT FACE xmin xmax ymin ymax zmin zmax` — selects the Faces totally within the specified bounding box
- `SELECT FACE -1 ibody1` — selects the Faces that are in common with a Face in `ibody1`
- `SELECT FACE -2 ibody1` — selects Faces that are within SolidBody `ibody1`
- `SELECT FACE $attrName1 attrValue1 ...` — selects the Faces that matches all the given Attributes

- `SELECT EDGE` — selects all Edges in selected Body
- `SELECT EDGE iedge` — selects Edge `iedge` in selected Body
  - using this is considered a bad practice since Edge numbering may change depending on the version of `OpenCASCADE` that is being used
- `SELECT EDGE ibody1 iford1 ibody2 iford2 iseq=1` — selects the Edge that has the indicated `ibody1/iford1`
  - as each Edge is created, it is marked with the `ibody/iford` of the Faces that adjoin it. This is the preferred technique.
- ...

- ...
- `SELECT EDGE xmin xmax ymin ymax zmin zmax` — selects the Edges totally within the specified bounding box
- `SELECT EDGE xmid ymid zmid` — selects the Edge whose midpoint is closest to the given coordinates
- `SELECT EDGE -1 ibody1` — selects the Edges that are in common with a Edge in `ibody1`
- `SELECT EDGE -2 ibody1` — selects Edges that are within SolidBody `ibody1`
- `SELECT EDGE $attrName1 attrValue1 ...` — selects the Edges that matches all the given Attributes

- `SELECT NODE` — selects all Nodes in selected Body
- `SELECT NODE inode` — selects Node `inode` in selected Body
  - using this is considered a bad practice since Node numbering may change depending on the version of OpenCASCADE that is being used
- `SELECT NODE x y z` — selects the Nodes closest to the given coordinates
- `SELECT NODE -1 ibody1` — selects the Nodes that are in common with a Node in `ibody1`
- `SELECT NODE -2 ibody1` — selects Nodes that are within SolidBody `ibody1`
- `SELECT NODE $attrName1 attrValue1 ...` — selects the Nodes that matches all the given Attributes

- `SELECT ADD ...` adds entities to the select list
- `SELECT SUB ...` subtracts (removes) entities from the select list
- `SELECT NOT` selects all entities not currently selected and unselects those that were selected
- `SELECT SORT key` sorts the select list by one of the keys: `$xmin`, `$ymin`, `$zmin`, `$xmax`, `$ymax`, `$zmax`, `$xcg`, `$ycg`, `$zcg`, `$area`, or `$length`

ATTRIBUTE density 2710 # global attribute

BOX 0 0 0 3 3 2

ATTRIBUTE tag \$block

BOX 1 1 0 1 2 2

ATTRIBUTE tag \$slot

SUBTRACT

CYLINDER -1 2 1 4 2 1 1/2

ATTRIBUTE tag \$hole

SUBTRACT

#-----

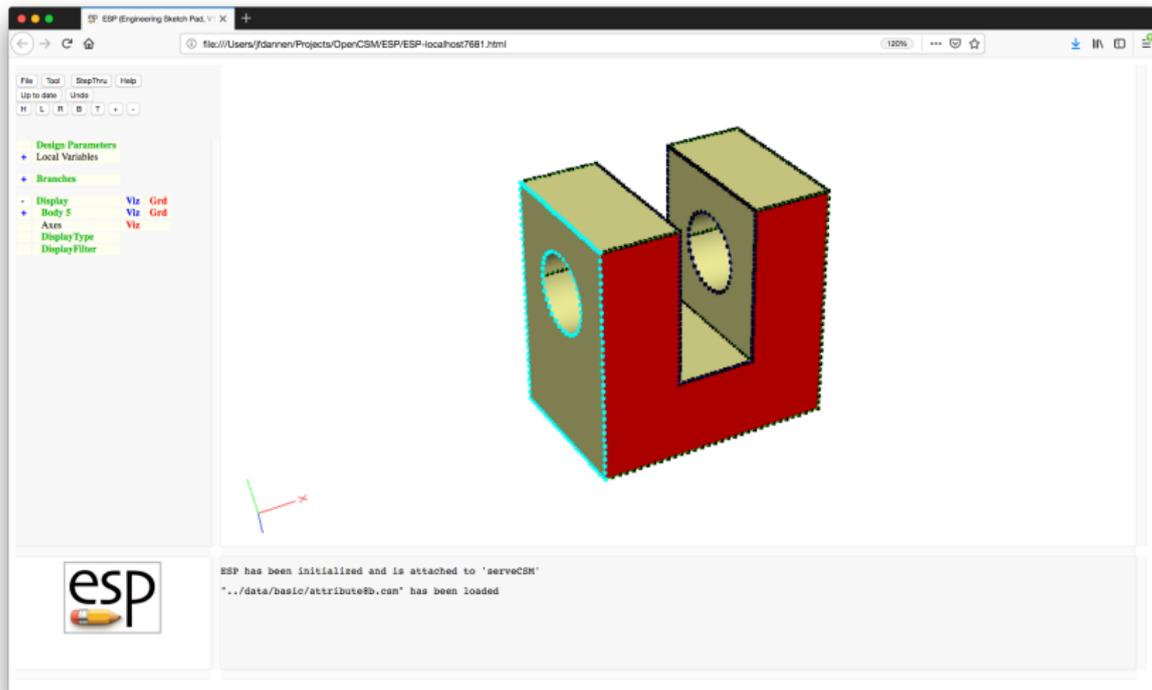
SELECT FACE 1 6 1 # select by FaceID

ATTRIBUTE \_color \$red

SELECT EDGE -.1 0.1 -.1 3.1 -.1 2.1 # select by bbox

ATTRIBUTE \_color \$cyan

ATTRIBUTE \_gcolor \$cyan



- Use `SELECT ADD ...` to add Faces, Edges, or Nodes to the selection list
- Use `SELECT SUB ...` to remove Faces, Edges, or Nodes from the selection List
- Both of these option use the selection type from the previous selection
  
- Use `SELECT SORT $key` to sort `@sellist` based upon `$xmin`, `$ymin`, `$zmin`, `$xmax`, `$ymax`, `$zmax`, `$xcg`, `$ycg`, `$zcg`, `$length` (if Edges), or `$area` (if Faces)



# Attributes Automatically Set to Bodys

<code>_body</code>	Body index (bias-1)
<code>_brch</code>	Branch index (bias-1)
<code>_tParams</code>	specified tessellation parameters: maximum side length, maximum specified sag, maximum angle
<code>_csys_*</code>	arguments when CSYSTEM was defined
<code>&lt;any&gt;</code>	all global attributes
<code>&lt;any&gt;</code>	all attributes associated with Branch that created Body
<code>&lt;any&gt;</code>	all attributes associated with "select \$body" statement

`_body` non-unique 2-tuple associated with first Face creation  
 [0] Body index in which Face first existed (bias-1)  
 [1] face-order associated with creation (see above)

`_brch` non-unique even-numbered list associated with Branches  
 that are active when the Face is created (most  
 recent Branch is listed first)  
 [2\*i ] Branch index (bias-1)  
 [2\*i+1] (see below)

Branches that contribute to `brch` attribute are

- `primitive` (for which `brch[2*i+1]` is face-order)
- `udprim.udc` (for which `brch[2*i+1]` is 1)
- `grown` (for which `brch[2*i+1]` is face-order)
- `applied` (for which `brch[2*i+1]` is face-order)
- `sketch` (for which `brch[2*i+1]` is Sketch primitive if  
 making WIRE)
- `patbeg` (for which `brch[2*i+1]` is pattern index)
- `recall` (for which `brch[2*i+1]` is 1)
- `restore` (for which `brch[2*i+1]` is Body number stored)



```

_body          non-unique 2-tuple associated with first Edge creation
  [0]          Body index in which Edge first existed (bias-1)
  [1]          100 * min(body[1][ileft],body[1][irite])
               + max(body[1][ileft],body[1][irite])
               (or -3 if non-manifold)

```

```

_edgeID        unique 5-tuple that is assigned automatically
  [0]          _faceID[0] of Face 1 (or 0 if non-manifold)
  [1]          _faceID[1] of Face 1 (or 0 if non-manifold)
  [2]          _faceID[0] of Face 2 (or 0 if non-manifold)
  [3]          _faceID[1] of Face 2 (or 0 if non-manifold)
  [4]          sequence number

```

...

...

```
_edgeID[0]/[1] swapped with edge[2]/[3]
  100*_edgeID[0]+_edgeID[1] > 100*_edgeID[2]+_edgeID[3]
if multiple Edges have same _edgeID[0], _edgeID[1],
  _edgeID[2], and _edgeID[3], then the sequence number
is defined based upon the first rule that applies:
* Edge with smaller xcg    has lower sequence number
* Edge with smaller ycg    has lower sequence number
* Edge with smaller zcg    has lower sequence number
* Edge with smaller length has lower sequence number
```

\_nface number of incident Faces

\_tParams specified tessellation parameters: maximum side length,  
maximum specified sag, maximum angle

<any> all attributes associated with "select \$edge" statement



# Attributes Automatically Set to Nodes

<code>_nodeID</code>	unique integer
<code>_nedge</code>	number of incident Edges
<code>&lt;any&gt;</code>	all attributes associated with "select \$node" statement



# Special User-defined Attributes for Bodys

- `_makeQuads` to make quads on all Faces in Body
- `_name` string used in ESP interface for a Body
- `_stlColor` color to use for all Faces in an `.stl` file



# Special User-defined Attributes for Faces

`_color`      color of front of Face in ESP  
              either R,G,B in three 0-1 reals  
              or \$red, \$green, \$blue, \$yellow, \$magenta,  
              \$cyan, \$white, or \$black

`_bcolor`      color of back of Face in ESP (see `_color`)

`_gcolor`      color of grid of Face in ESP (see `_color`)

`_makeQuads`   to make quads for this Face

`_stlColor`    color to use for this Face in an `.stl` file



# Special User-defined Attributes for Edges

`_color`      color of front of Edge in ESP  
                 either R,G,B in three 0-1 reals  
                 or \$red, \$green, \$blue, \$yellow, \$magenta,  
                 \$cyan, \$white, or \$black

`_gcolor`      color of grid of Edge in ESP (see `_color`)



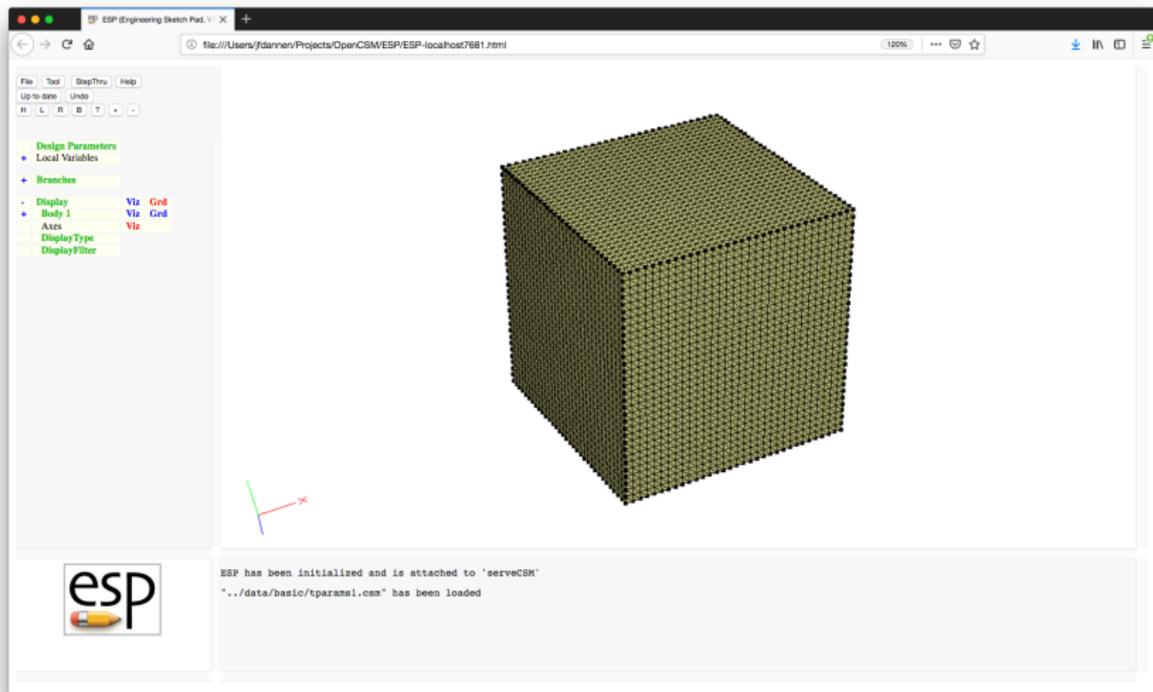
# Special User-defined Attributes for Nodes

`_color`      color of Node in ESP  
either R,G,B in three 0-1 reals  
or \$red, \$green, \$blue, \$yellow, \$magenta,  
\$cyan, \$white, or \$black

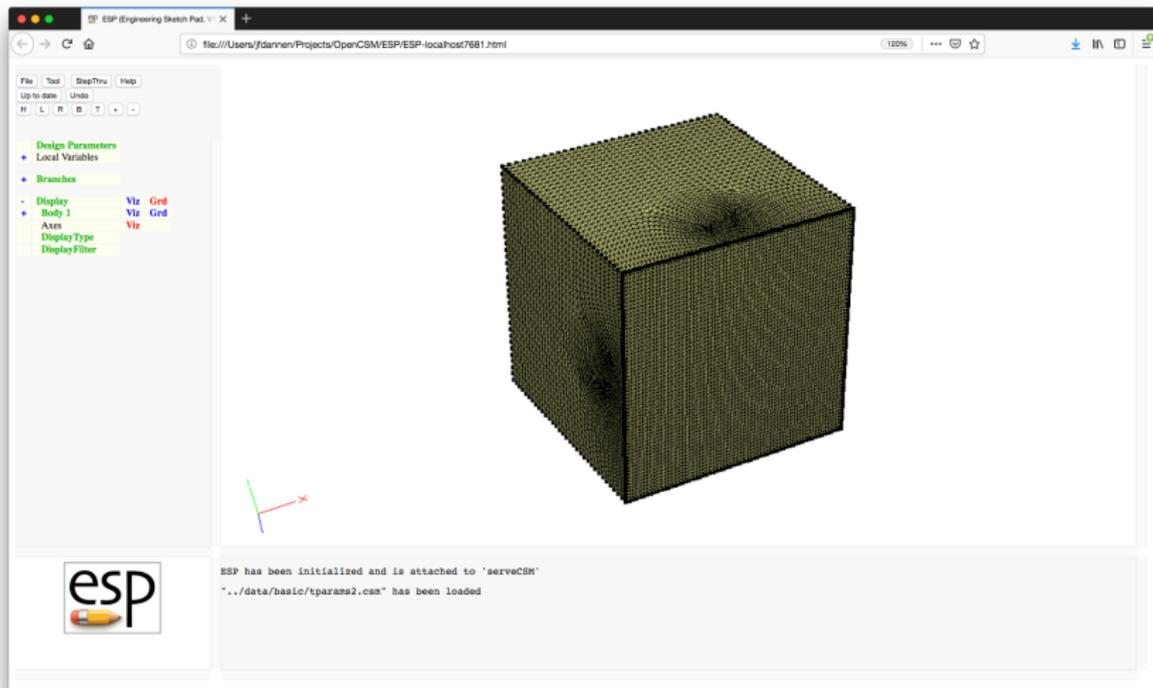


# Tessellation Parameters (1)

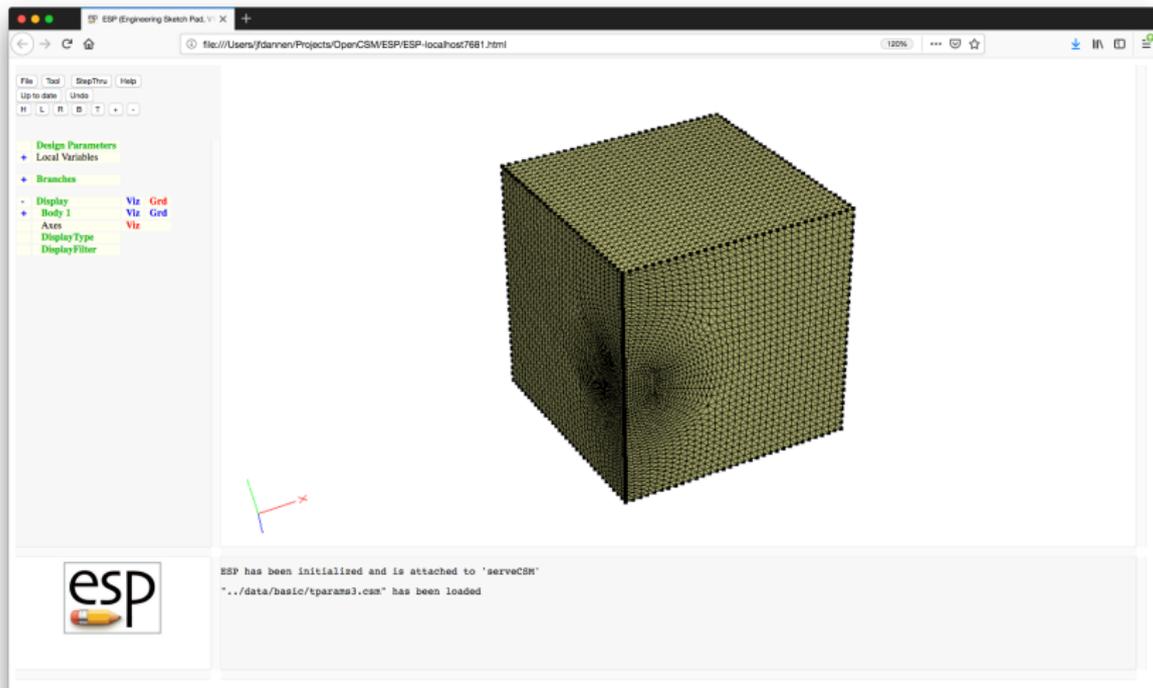
Default tessellation has `_tParams = 0.043; 0.013; 20`



```
SELECT FACE 1 6  
ATTRIBUTE .tParams "0.02; 0.013; 20"
```



```
SELECT    EDGE 1 1 1 6
ATTRIBUTE .tParams "0.02; 0.013; 20"
```



- Csystems (coordinate systems) are generated by the **CSYSTEM** statement and are applied to the Body on the top of the Stack
- Csystems are treated in many ways like Attributes
  - Csystem names must not be the same as an Attribute name
  - Csystems are found in **ESP** in same place as Attributes
- Csystems are transformed along with any transformations that are applied to their Body

- Format of the CSYSTEM statement is:
  - If argument to CSYSTEM contains 9 entries:  
`{x0, y0, z0, dx1, dy1, dz1, dx2, dy2, dz2}`  
origin is at `(x0,y0,z0)`  
`dirn1` is in `(dx1,dy1,dz1)` direction  
`dirn2` is in `(dx2,dy2,dz2)` direction
  - If argument to CSYSTEM contains 5 entries and first is positive:  
`{+iface, ubar0, vbar0, du2, dv2}`  
origin is at normalized `(ubar0,vbar0)` in `iface`  
`dirn1` is normal to Face  
`dirn2` is in `(du2,dv2)` direction

- Format of the CSYSTEM statement is:
  - If argument to CSYSTEM contains 5 entries and first is negative:

```
{-iedge, tbar, dx2, dy2, dz2}
```

origin is at normalized (tbar) in iedge

dirn1 is tangent to Edge

dirn2 is part of (dx2,dy2,dz2) that is  
orthogonal to dirn1

- If argument to CSYSTEM contains 7 entries:

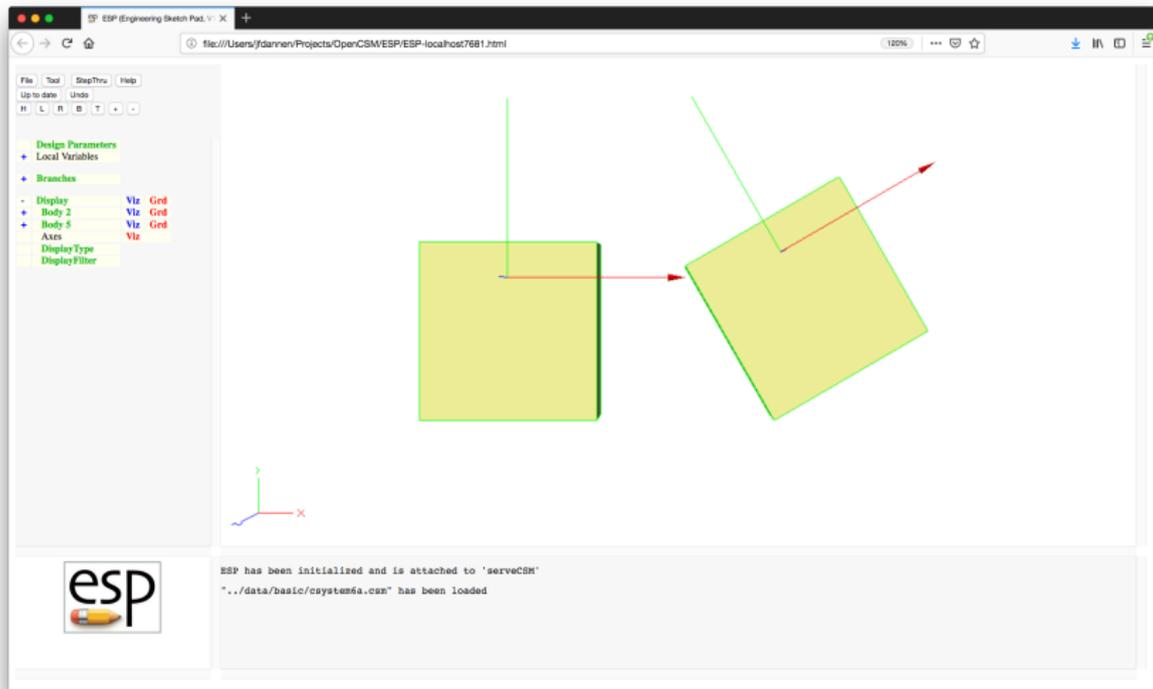
```
{inode, dx1, dy1, dz1, dx2, dy2, dz2}
```

origin is at Node inode

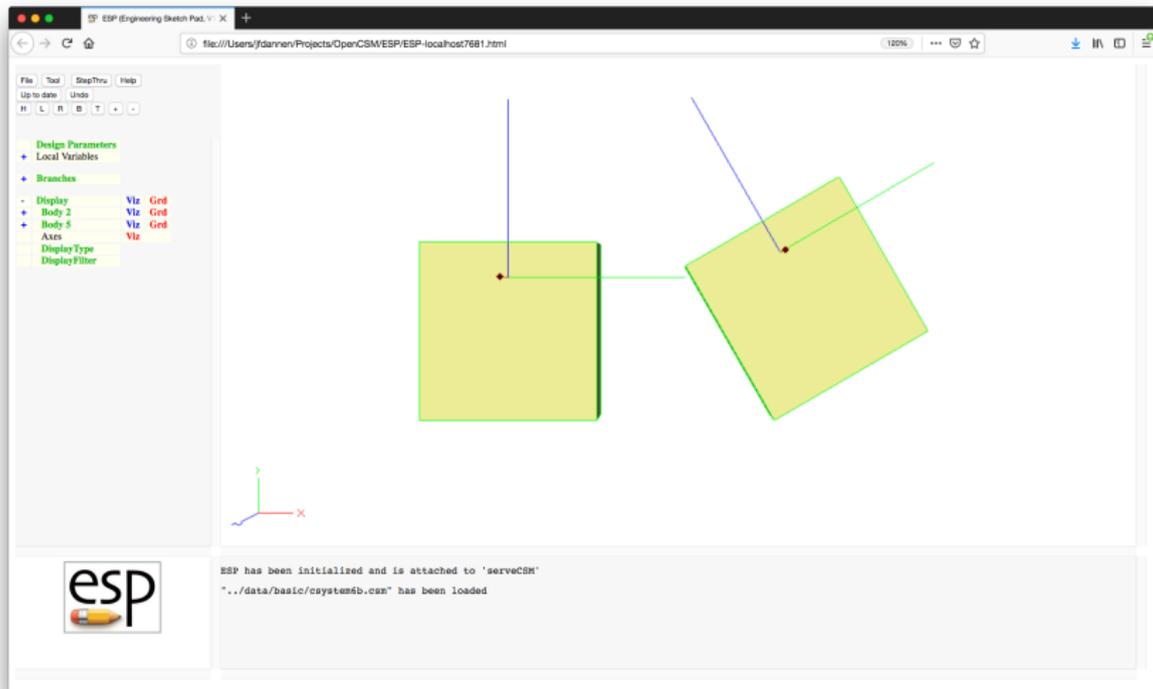
dirn1 is in (dx1,dy1,dz1) direction

dirn2 is part of (dx1,dy2,dz2) that is  
orthogonal to dirn1

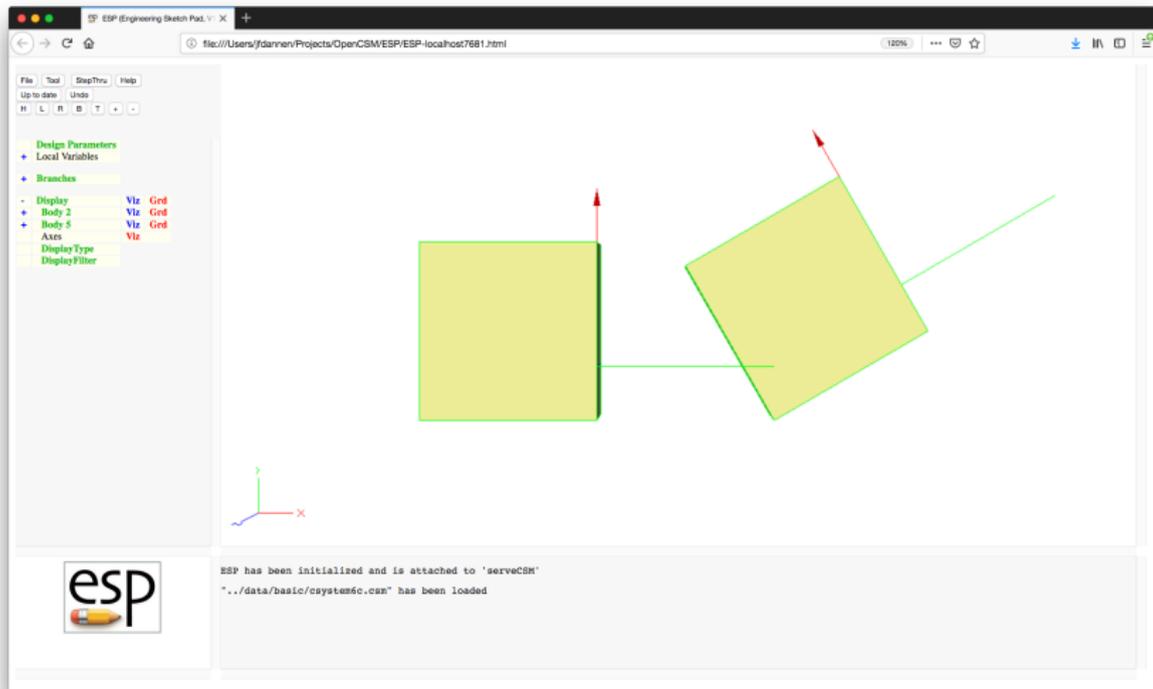
```
CSYSTEM cs1 "0.5; 0.8; 1.1; 1;0;0; 0;1;0"
```



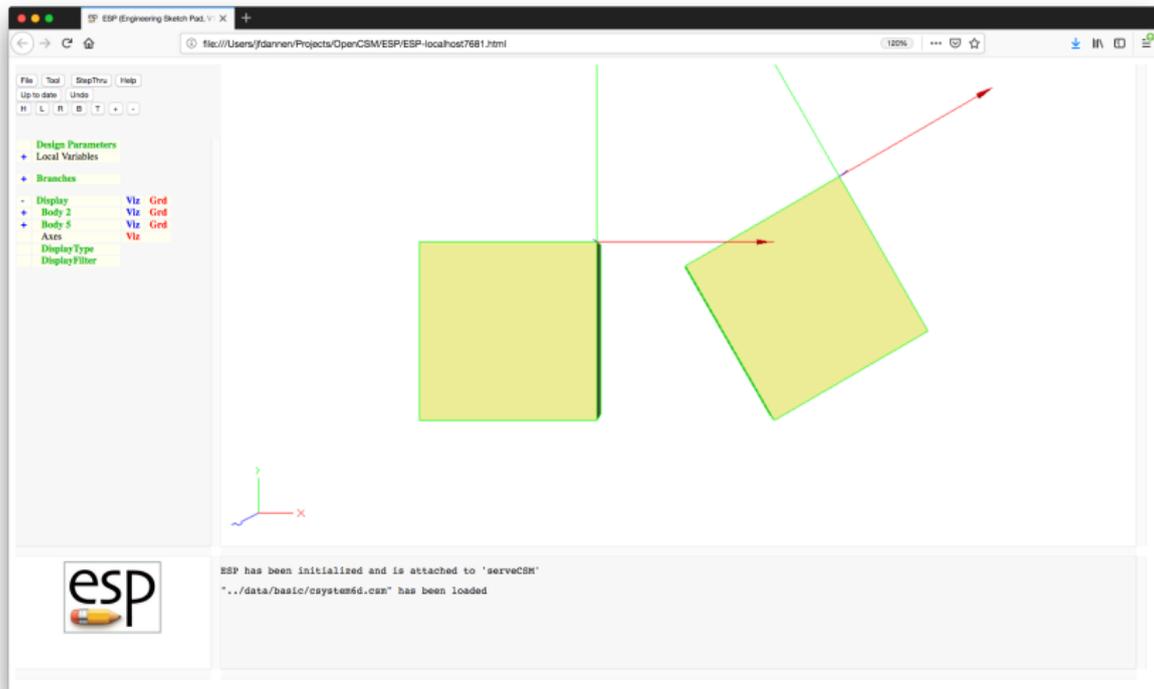
```
CSYSTEM cs1 "+6; 0.5; 0.8; 1;0"
```



```
CSYSTEM cs1 "-6; 0.3; 1;0;0"
```



```
CSYSTEM cs1 "7;    1;0;0;    0;1;0"
```



The screenshot shows the ESP Engineering Sketch Pad interface. The main workspace displays two rectangular bodies. The left body is upright, and the right body is tilted. Red arrows indicate a relationship between the top edges of the two bodies. A coordinate system is visible in the bottom left of the sketch area.

On the left side, there is a sidebar with the following sections:

- Design Parameters
  - Local Variables
- Branches
  - Display Viz Grid
  - Body 2 Viz Grid
  - Body 5 Viz Grid
  - Axes Viz
  - Display Type
  - Display Filter

At the bottom of the interface, there is a status bar with the ESP logo and the following text:

```
ESP has been initialized and is attached to 'serveCSM'
'../data/basic/csystem6d.csm' has been loaded
```



# Attribute Editor (1)

- Best practice is to set Attributes when entity is first created
- If not possible, the `editAttr` UDF is available to set Attributes based upon the Attributes of an entity's neighbors

- Statements in the attribute editor can be one of:
  - NODE      <selector> <attrName1=attrValue1> ...
  - EDGE      <selector> <attrName1=attrValue1> ...
  - FACE      <selector> <attrName1=attrValue1> ...
  - AND       <selector> <attrName1=attrValue1> ...
  - ANDNOT <selector> <attrName1=attrValue1> ...
  - SET                      <attrName1=attrValue1> ...
- Keywords can either be specified in lowercase or UPPERCASE
- <selector> can be one of HAS, ADJ2NODE, ADJ2EDGE or ADJ2FACE

- Typical block of code looks like:

```
NODE ADJ2FACE tagType=spar tagIndex=1
AND  ADJ2FACE tagType=lower
AND  ADJ2EDGE tagType=root
SET           capsConstraint=pointConstraint1
```

- Patterns can be used with PATBEG and PATEND

```
# SolidBody
BOX          0 -1 -1  3  2  2
ATTRIBUTE type $OML
SELECT      FACE @nbody  1
ATTRIBUTE face $xmin
SELECT      FACE @nbody  2
ATTRIBUTE face $xmax
SELECT      FACE @nbody  3
ATTRIBUTE face $ymin
SELECT      FACE @nbody  4
ATTRIBUTE face $ymax
SELECT      FACE @nbody  5
ATTRIBUTE face $zmin
SELECT      FACE @nbody  6
ATTRIBUTE face $zmax
STORE       SolidBody
```

```
# get bounding box of SolidBody
RESTORE    SolidBody
SET       xmin    @xmin
SET       xmax    @xmax
SET       ymin    @ymin
SET       ymax    @ymax
SET       zmin    @zmin
SET       zmax    @zmax
STORE    .
```

```
# Waffle (centered on SolidBody)
UDPRIM    waffle    filename <<    depth zmax-zmin+2
  POINT   A AT    xmin-1  (ymin+ymax)/2
  POINT   B AT    xmax+1  (ymin+ymax)/2
  LINE    AB  A   B   type=symmetry

  PATBEG  i   3
    POINT  C AT    xmin+i/4*(xmax-xmin) ymin-1
    POINT  D AT    xmin+i/4*(xmax-xmin) ymax+1
    LINE   .   C   D   bulkhead=!val2str(i,0)
  PATEND

>>
TRANSLATE 0 0  zmin-1
STORE     Waffle
```

```
# score the SolidBody by the Waffle and extract Faces
RESTORE    SolidBody
RESTORE    Waffle
SUBTRACT
EXTRACT    0

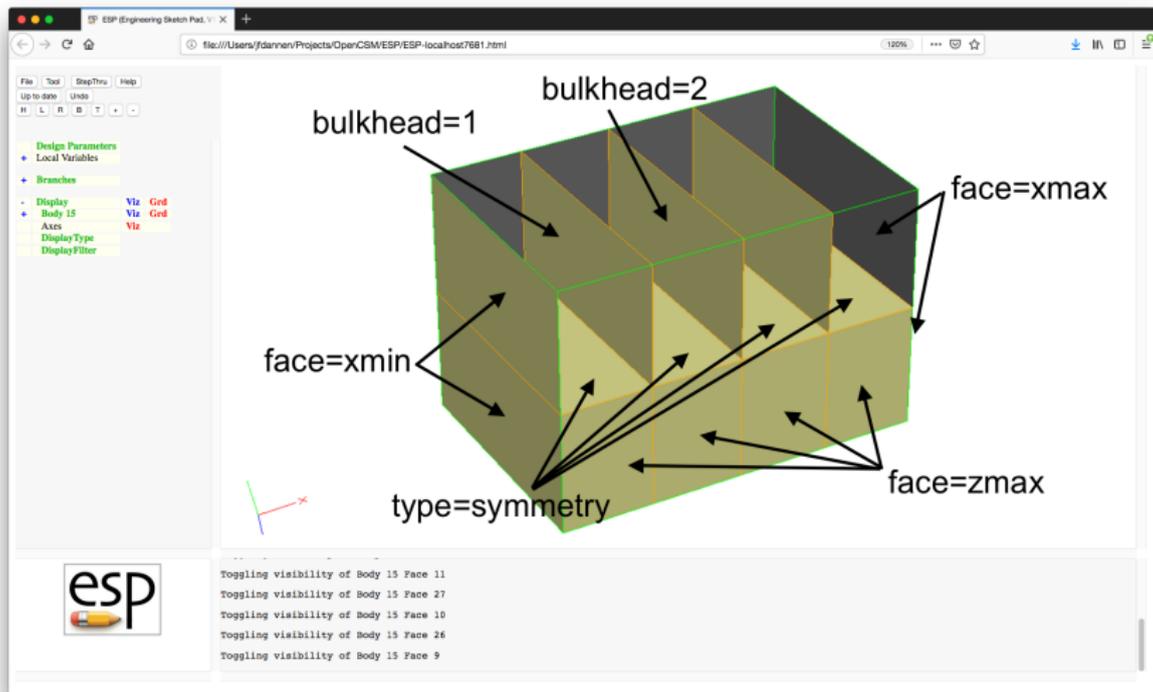
# generate the internal structure
RESTORE    SolidBody
RESTORE    Waffle
INTERSECT

# put them together
UNION
```



# EditAttr Example (5)

Some Faces not shown for clarity





# EditAttr Example (6)

Attributes that we want to define

The screenshot shows the ESP (Engineering Sketch Pad) interface. The main window displays a 3D model of a cube with a grid on its top face. Three annotations with arrows point to specific features: 'found=face' points to the top face, 'found=node' points to a corner vertex, and 'found=edge' points to a dashed line representing an edge. The left sidebar contains a tree view with categories like Design Parameters, Local Variables, Branches, and Display. The bottom status bar shows the ESP logo and a list of actions: 'Toggling visibility of Body 15 Face 10', 'Toggling visibility of Body 15 Face 26', 'Toggling visibility of Body 15 Face 9', 'Press "Stepthru" to enter Stepthru mode', and 'Toggling grid of Body 15 Face 31'.

```
# select Face on bulkhead=1 but top half
UDPRIM    editAttr  filename  <<
  FACE    HAS        bulkhead=1
  AND     ADJ2FACE   face=ymax
  SET                                found=face
>>
```

```
# select Edge on OML seam between bulkheads 1 and 2
UDPRIM      editAttr  filename  <<
  NODE      ADJ2FACE  bulkhead=1
  SET                               bulkhead=1
  NODE      ADJ2FACE  bulkhead=2
  SET                               bulkhead=2
  EDGE      ADJ2FACE  face=zmax
  AND        ADJ2FACE  type=symmetry
  AND        ADJ2NODE  bulkhead=1
  AND        ADJ2NODE  bulkhead=2
  SET                               found=edge
  NODE      HAS        bulkhead=1
  SET                               bulkhead=
  NODE      HAS        bulkhead=2
  SET                               bulkhead=
>>
```

```
# select Node on OML seam at the outlet
UDPRIM      editAttr  filename  <<
  NODE      ADJ2FACE  face=xmax
  AND       ADJ2FACE  face=zmax
  AND       ADJ2FACE  type=symmetry
  SET
  >>
```

- Using  
`$ESP_ROOT/training/ESP/data/session08/wingStruct.csm`
  - put the Attribute `LoadPoint=leftTip` on the Node that is at the intersection of the forward spar, wing tip, and upper skin on the left wing
  - for the skin panels on the right wing that are between the first and second rib, make their color red and their grid white
  - make the Edges blue that are between two red panels