

Engineering Sketch Pad (ESP)



Training Session 2.2 Attribution

John F. Dannenhoffer, III

jfdannen@syr.edu
Syracuse University

Bob Haimes

haimes@mit.edu

Massachusetts Institute of Technology

Marshall Galbraith

galbramc@mit.edu

- Purpose and Types of Attributes
- Selecting Entities
- Setting Attributes
 - UDPRIM `editAttr`
- Viewing Attributes
 - DisplayFilter

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

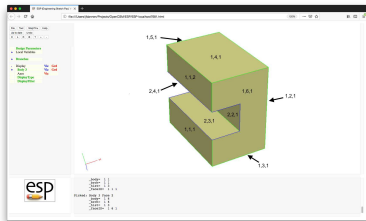
- Attributes are defined for any Branch that produces a Body
- Attributes are defined by an **ATTRIBUTE** statement
- Attribute names must not start with a period (which is reserved for EGADS) or an underscore (which is reserved for OpenCSM)
- If the first character of the value is a dollar-sign, then 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)

- Best practice is to set the Attributes as soon as the Body is created (for example via a primitive or grown Body command)
- Attributes can be set at any time via an `ATTRIBUTE` statement associated with a `SELECT` statement

- 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 $attrName1 $attrValue1 ...` — selects the last Body that matches all the given (string) 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 \$attrName1 \$attrValue1 ...** – selects the Faces that matches all the given (string) Attributes



iford

BOX	0	0	0	2	3	3
-----	---	---	---	---	---	---

BOX	-1	1	-1	2	1	5
-----	----	---	----	---	---	---

SUBTRACT

END

- Edge ID is generated based upon the ibody/iface of its two adjoining Faces

- `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 $attrName1 $attrValue1 ...` – selects the Edges that matches all the given (string) 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 $attrName1 $attrValue1 ...` – selects the Nodes that matches all the given (string) Attributes

- 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

EGADS attributes assigned to Bodys:

<code>_body</code>	Body index (bias-1)
<code>_brch</code>	Branch index (bias-1)
<code>_tParams</code>	tessellation parameters
<code>_csys_*</code>	arguments when CSYSTEM was defined
<code><any></code>	all global attributes
<code><any></code>	all attributes associated with Branch that created Body
<code><any></code>	all attributes associated with "select \$body" statement

EGADS attributes assigned to Faces:

<code>_body</code>	non-unique 2-tuple associated with first Face creation
<code>[0]</code>	Body index in which Face first existed (bias-1)
<code>[1]</code>	face-order associated with creation (see above)
<code>_brch</code>	non-unique even-numbered list associated with Branches that are active when the Face is created (most recent Branch is listed first)
<code>[2*i]</code>	Branch index (bias-1)
<code>[2*i+1]</code>	(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)

`_faceID` unique 3-tuple that is assigned automatically
 `[0]` `body[0]`
 `[1]` `body[1]`
 `[2]` sequence number

if multiple Faces have same `_faceID[0]` and `_faceID[1]`,
then the sequence number is defined based upon the
first rule that applies:

- * Face with smaller `xcg` has lower sequence number
- * Face with smaller `ycg` has lower sequence number
- * Face with smaller `zcg` has lower sequence number
- * Face with smaller area has lower sequence number

`_hist` list of Bodies that contained this Face (oldest to newest)
<any> all attributes associated with Branch that first created Face
<any> all attributes associated with "SELECT \$face" statement

EGADS attributes assigned to Edges:

<code>_body</code>	non-unique 2-tuple associated with first Edge creation
<code>[0]</code>	Body index in which Edge first existed (bias-1)
<code>[1]</code>	$100 * \min(\text{body}[1][\text{ileft}], \text{body}[1][\text{irite}])$ $+ \max(\text{body}[1][\text{ileft}], \text{body}[1][\text{irite}])$ (or -3 if non-manifold)
<code>_edgeID</code>	unique 5-tuple that is assigned automatically
<code>[0]</code>	<code>_faceID[0]</code> of Face 1 (or 0 if non-manifold)
<code>[1]</code>	<code>_faceID[1]</code> of Face 1 (or 0 if non-manifold)
<code>[2]</code>	<code>_faceID[0]</code> of Face 2 (or 0 if non-manifold)
<code>[3]</code>	<code>_faceID[1]</code> of Face 2 (or 0 if non-manifold)
<code>[4]</code>	sequence number

...

EGADS attributes assigned to Edges:

...

```
_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

<any>

all attributes associated with "select \$edge" statement

EGADS attributes assigned to Nodes:

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

- 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, dz3}`
origin is at `(x0,y0,q0)`
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



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
```


- 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)

- Using `$ESP_ROOT/training/session2.2/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