

## Mesh Controller

How to use:

Filtering methods:

Geometry:

Pick Object:

Reference Object:

Use Sub-Object Selection:

Use methods of :

Search in:

Vertex:

Scalar Output:

Vector Output:

Matrix outputs:

Edge

Scalar Output:

Vector Output:

Matrix outputs:

Face

Scalar Output:

Vector Output:

Matrix outputs:

Object

Scalar Output:

Vector Output:

Matrix output:

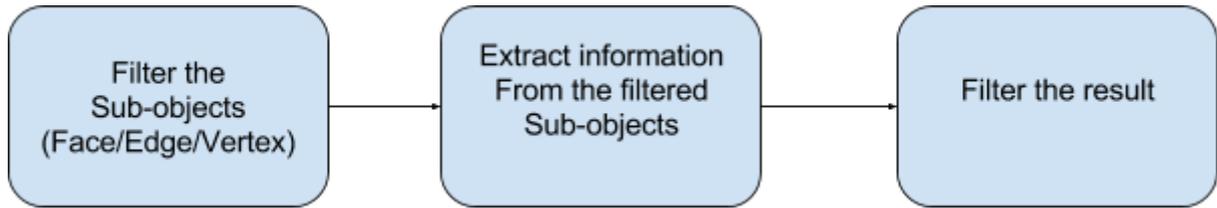


### **Mesh Controller**

This controller extract information about Faces,Edges and Vertices of an Editable Mesh or Editable Poly object.

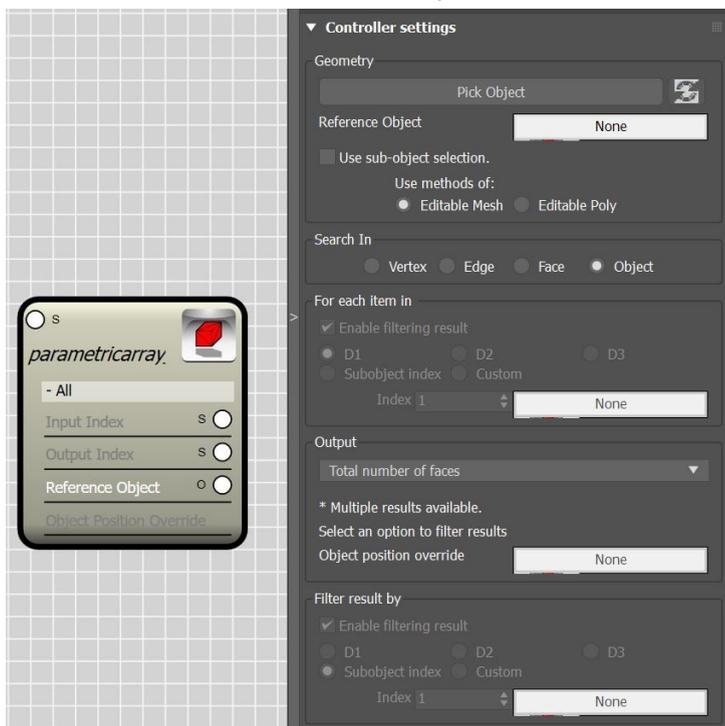
How to use:

There are three steps to be taken when using Mesh controller.

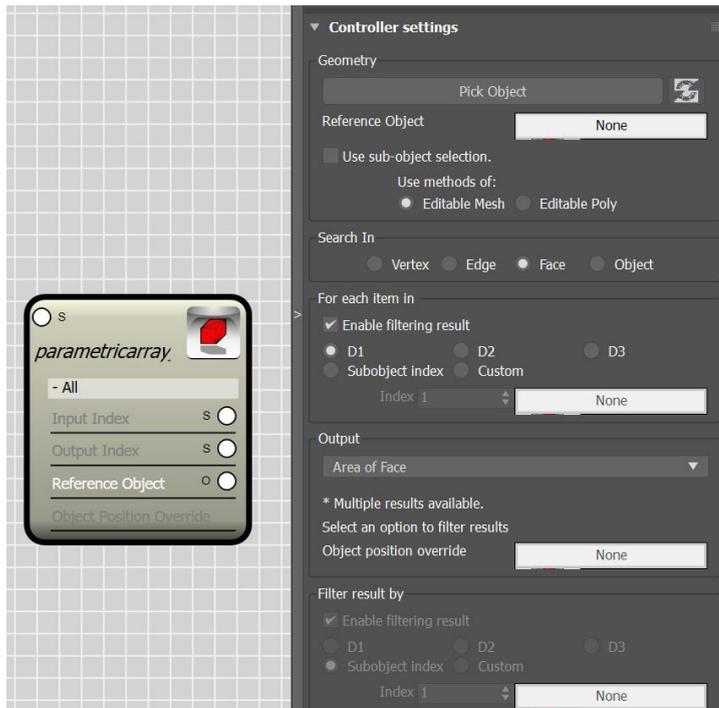


For example to retrieve the face center of a given face , you must choose “Face” from sub-object list, then choose “Center point of the face” from output menu. Since each face has only one center the result will be single value and it does not require filtering. But if you wish to find the vertex position of the given face then you must choose “Face” from sub-object list and “\*Actual vertex position” from the output menu. Notice that each face may have more than three vertices, therefore we must select one of the vertices by filtering the result using options available in this section. If you don’t choose any filter, the controller will return an array of position vectors. See below different scenarios and more information:

1. The area of search does not require filtering the inputs. For example choosing “Object” for the searching area you will have single outputs. Therefore the input and output filtering is disabled. Example below shows how to retrieve the number of faces from the editable mesh object.

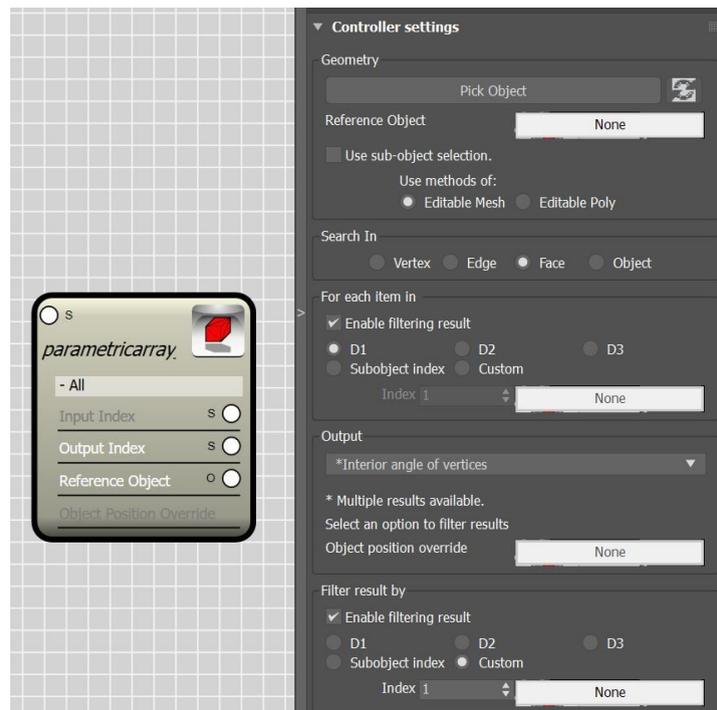


2. The area of search requires filtering. When choosing Faces/Edges or Vertices for searching area you have two options to proceed.
  - 2.1. Enable filtering the inputs and select a particular sub-object from the object. In next step depending on which option is selected from output list you will have different options to proceed:
    - 2.1.1. Selected option returns a single value. For example choosing the Face area as an output does not require further filtering. See below image.



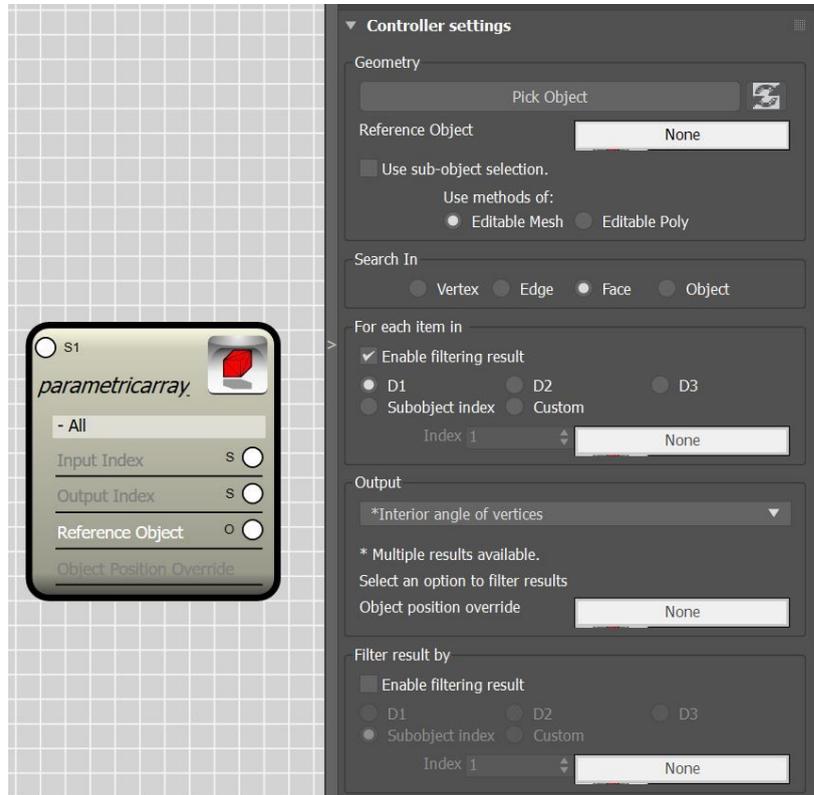
2.1.2. Selected option returns an array. In this case you may enable the filtering for output values or not. Note the options that start with (\*) will return multiple results.

2.1.2.1. If “Enable filtering results” is selected then a particular output will be return. The controller output will remain single. In below example the controller return the interior angle for the first vertex in the face. Note the filter is set to custom and index is equal to 1.



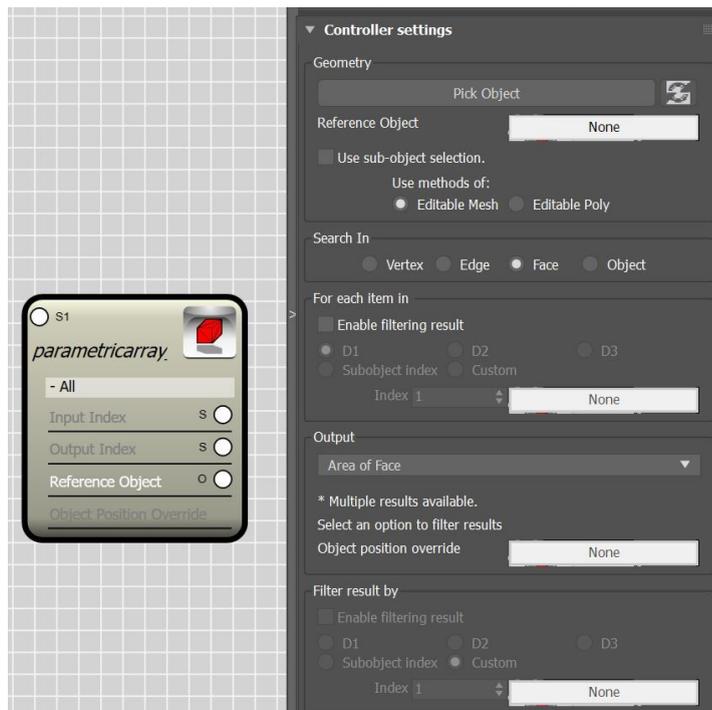
2.1.2.2. If “Enable filtering results” is not selected result will be returned in array format. The controller output becomes a 1D list. In

example below the interior angle of all vertices of one face will be return in a 1 dimensional array (S1).



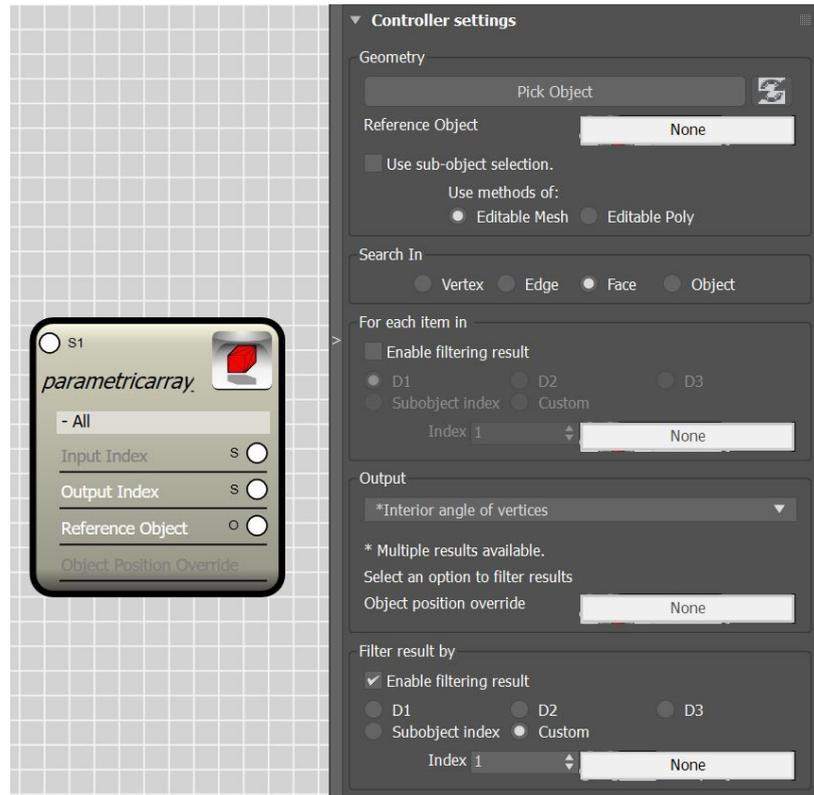
2.2. Disable the filtering the inputs will push all the results in an array and depend on what you choose from output may produce different result.

2.2.1. If selected output returns single output then controller will return a one dimensional array which contains the result for all sub-objects in search area. For example in below you can retrieve the face area of all faces in editable mesh object in one array (S1).



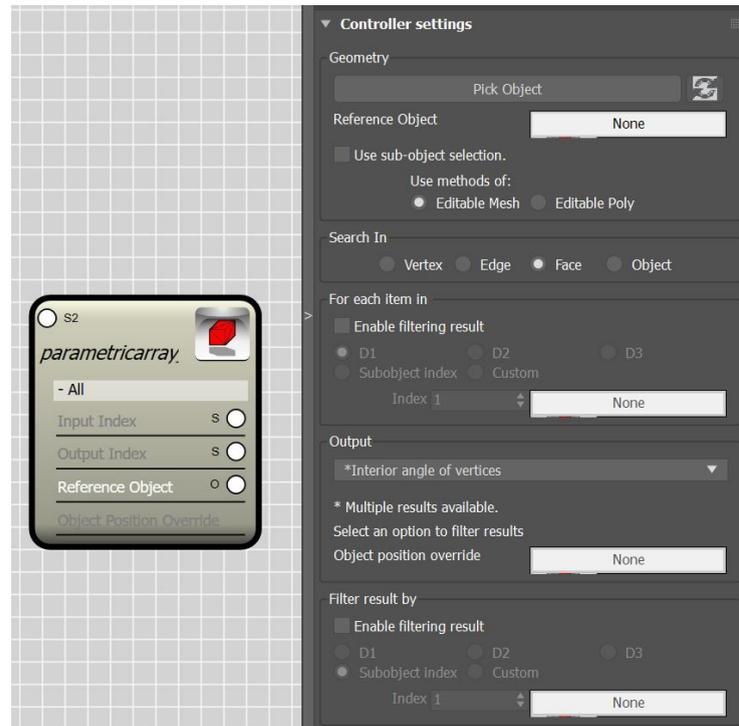
2.2.2. If selected output returns multiple results (starts with stark mark) then depending on filtering result or not filtering result we can proceed with below options:

2.2.2.1. If filtering output is enabled then 1 dimensional array will be returned. For example you can collect the interior angles of the first vertex in all faces of the editable mesh in one array like the image below (S1)



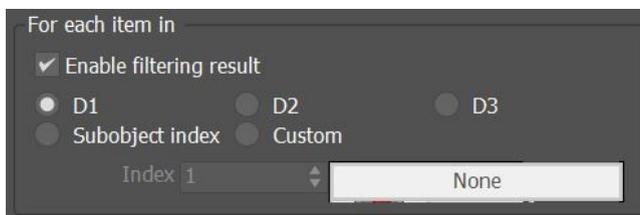
2.2.2.2. If filtering output is not enabled then all the result will be collected in array format for each subobject in the node. This will result in a nested array or simply a 2 dimensional array. For example in below controller returns the interior angles of all vertices of all faces of the editable mesh in a 2 dimensional

array.(S2)



In above example if an Hedra object is selected as Editable-Mesh the result will be a 2D array like:  
#(#(60.0, 60.0, 60.0), #(60.0, 60.0, 60.0), #(60.0, 60.0, 60.0), #(60.0, 60.0, 60.0), #(60.0, 60.0, 60.0), #(60.0, 60.0, 60.0), #(60.0, 60.0, 60.0), #(60.0, 60.0, 60.0))  
An array of 8 array which contains the interior angles in each face, in the Hedra all the interior angles are 60 degree.

### Filtering methods:



Filters the inputs (Vertices/Edges or Faces) or outputs using an Element Identifier. The inputs or outputs are in form of arrays that each member can be access using an integer value called Element Identifier or simply an ID. Let's assume the input or output array is formed and saved in a variable name **A**. The result of filtering will be simply to retrieve the element from the array **A** by a specific **ID** : **A[ID]**  
There are different ways of defining the **ID**.

1. **D1 /D2 and D3**

The corresponding index value in each dimension will be considered as ID. If we assume that **A** is an array of inputs (Vertices/Edges or Faces) or outputs (vertex positions, edge endpoints or face vertices) :

D1  $\Rightarrow$  use index1 as ID to pick the item  $\Rightarrow A[Index1]$

D2  $\Rightarrow$  use index1 as ID to pick the item  $\Rightarrow A[Index2]$

D3  $\Rightarrow$  use index1 as ID to pick the item  $\Rightarrow A[Index3]$

## 2. Subobject Index

Filter the items by the subobject index of the controller. If mesh controller is assigned to a property which contains a subobject index parameter (for example the vertex position of a spline shape) then the index of the subobject will be used to pick the item from the array. Assume the Mesh controller is assigned to vertex number 3 of a container which contains spline shape with 5 vertices. In this case the subobject index value is 3.

Subobject Index  $\Rightarrow$  use subobject index as ID to pick the item  $\Rightarrow A[i]$

## 3. Custom

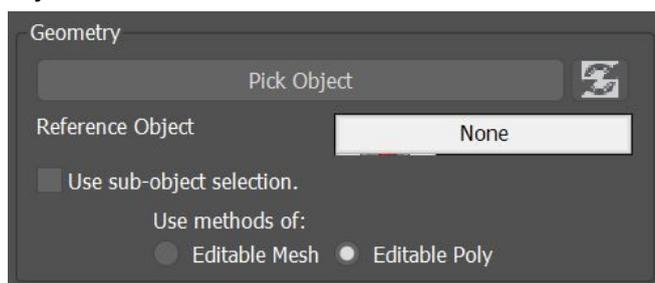
You can specify the ID either by setting the index value from control panel or assigning a controller to the relevant channel.

### Note:

- In all above cases if the ID is greater than the array size then the last element will be returned.
- If the ID is less than 1 then the first element will be returned.
- If provided ID is not an integer then it will be converted to integer using **int** method.

### Geometry:

Mesh controller accepts both Editable Mesh and Editable Poly object. However the methods that are used for these types are different and may produce incorrect result if wrong type of object has been selected.



### Pick Object:

You can select the object directly from the scene using the pick object button. Once the object is selected the name of the object will be shown on the button and the suitable methods will be automatically set.

### Reference Object:

You may pass an editable poly or editable mesh using another controller (for example Para link-controller). In this case you must set the methods manually.

Use Sub-Object Selection:

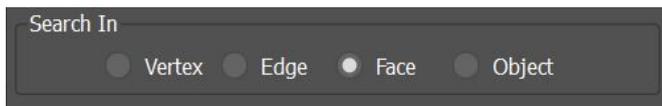
The controller works only on selected sub-objects (Face,Edges or Vertices) of the given object and ignores the rest of sub-object elements.

Use methods of :

You must selected the appropriate type of the object which you select or you pass by a controller. If the object is an editable Mesh ( any object that can be converted to mesh) then use Mesh methods. If the object is an Editable Poly (objects with Edit-poly modifier on top or and editable poly object) then you can use Editable Poly methods.

**Important Note: Incorrect method may not cause any error but it may produce incorrect result. Editable mesh methods return the vector results in local coordinate system of the node while Editable poly methods work in world coordinate system.**

Search in:



Defines the sub-object level which the poly/mesh object must be explored. Each option provides with separate list of outputs in output menu.

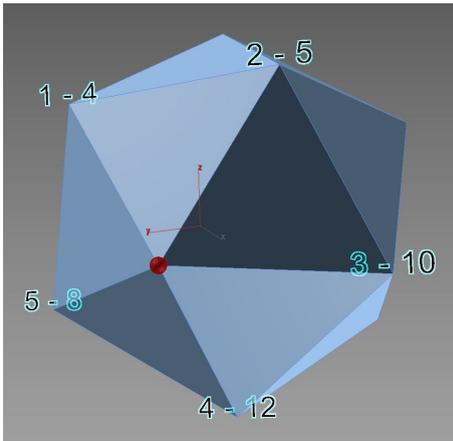
Vertex:

Set search area to the vertex level.

*Scalar Output:*

When scalar is selected following options are available for the output:

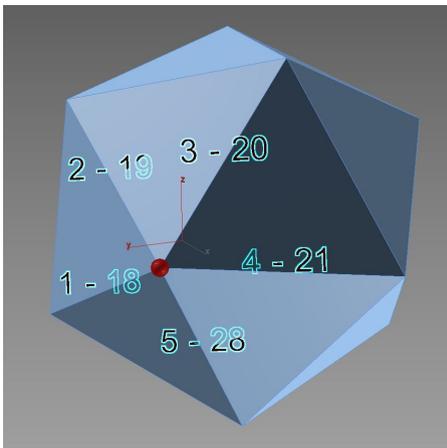
1. Number of edges connected to vertex  
Note that for editable meshes the hidden edges are calculated
2. Number of faces contained in vertex  
Number of faces that shares this particular vertex. In editable meshes counts the triangulated faces.
3. Total angle of vertices  
A planar closed vertex will return 360
4. Minimum distance from neighboring vertex  
Returns the length of shortest edge connecting this vertex to other
5. Gaussian Curvature  
Returns the [Gaussian Curvature](#) of the mesh surface at the vertex location.
6. Mean Curvature  
Returns the [Mean Curvature](#) of the mesh surface at the vertex location.
7. \*Index of neighboring vertices (ordered)  
Returns the index of the petal vertices . the array of indexes are in order ( it can be clockwise or anti-clockwise)



In Above example the petal vertices of selected vertex are 4,5,10,12,8

8. \*Index of edges connected to this vertex (ordered)

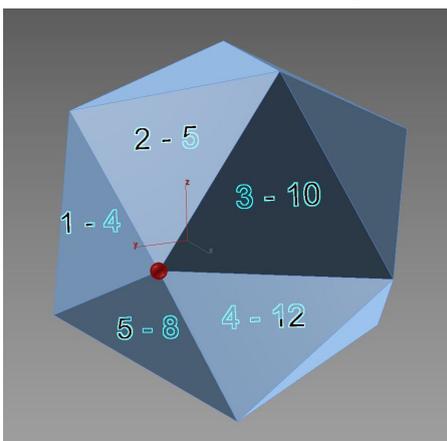
Returns the index of the petal edges. the array of indexes are in order ( it can be clockwise or anti-clockwise)



In Above example the petal vertices of selected vertex are 18,19,20,21,28, note this is an editable poly, for editable mesh there are two edges connecting two vertices , see more information on MAX SCRIPT documentation.

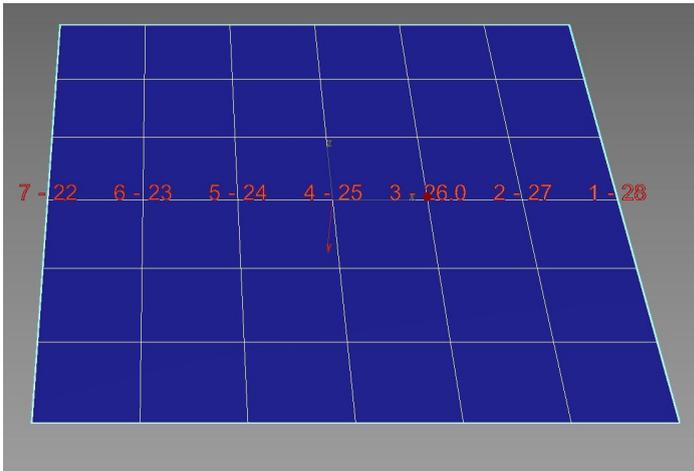
9. \*Index of faces contained in vertex (ordered)

Returns the index of the petal fales. the array of indexes are in order ( it can be clockwise or anti-clockwise)



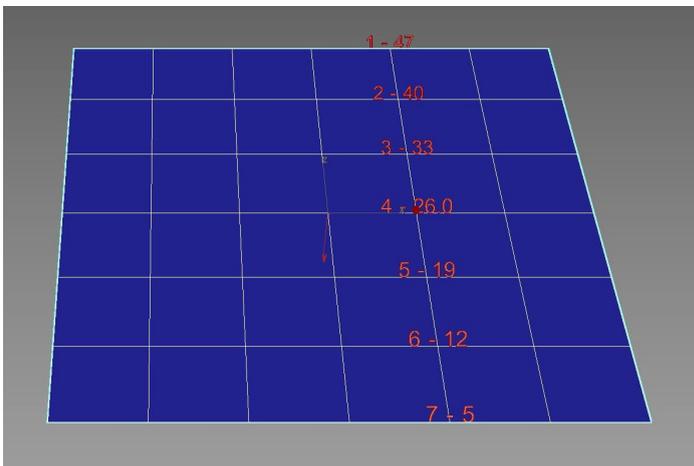
In above example the petal faces of the given vertex are 4,5,10,12,8

10. \*Index of vertices in same ring



In above example the vertices 28,27,26,25,24,23, and 22 are in same ring with vertex 26. Note the controller returns the given vertex index as part of the ring.

11. \*Index of vertices in same Loop



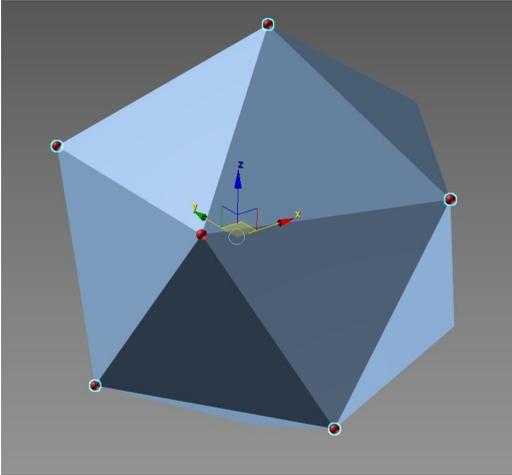
In Above example vertices 47,40,33,26,19,12,5 are in same loop with vertex 26 , Note the controller returns the given vertex index as part of the loop.

*Vector Output:*

When vector is selected following options are available for the output:

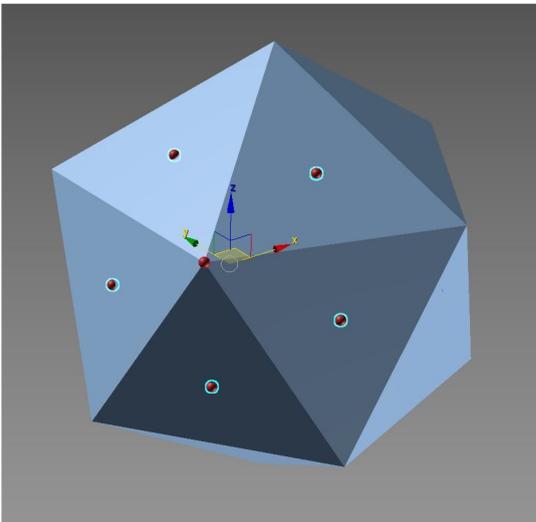
1. Actual vertex position  
Note that if the controller uses Edit-poly mode the position will be calculated in world coordinate system,however, if Edit-Mesh mode is selected the position will be returned in local coordinate system of the node.
2. Normal vector at vertex location  
Note that for mesh object normal vector is computed using *getNormal* method of MAX SCRIPT while for Editable Poly object the normal vector of the vertex is the vector sum of the normals of the adjacent faces.
3. \*Position of neighboring vertices

Returns the position of petal vertices. See the notes for [Actual vertex position](#)



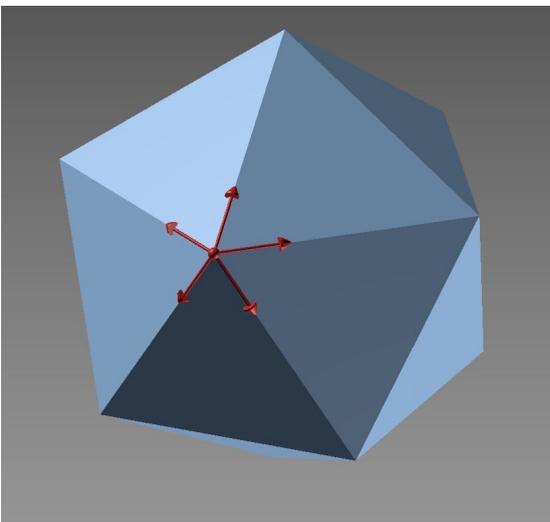
4. \*Center of adjacent faces

Returns the center of petal faces. Note that the result for Editable mesh objects is calculated in local coordinate system



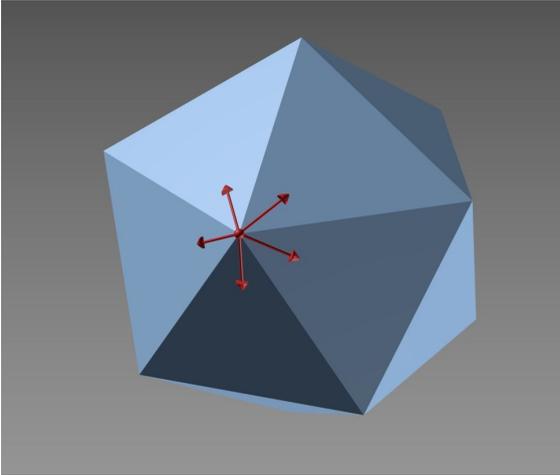
5. \*Outgoing edge as vector

Returns the direction of petal edges (normalized)



6. \*Vertex bisector

Returns the bisector of the interior angles.



#### *Matrix outputs:*

When Matrix is selected the controller builds a transformation matrix at given vertex location and Z axis will be aligned with outgoing edges.

#### Edge

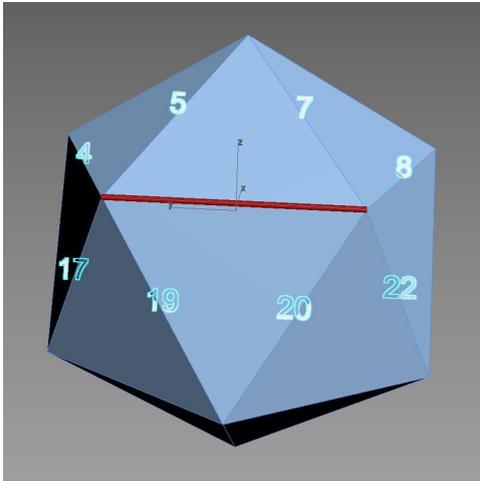
Set search area to the Edge level. Note that in Editable mesh objects there are two edges in same place while in Editpoly objects only edge connects two faces. For more information see Max-Script documentation.

#### *Scalar Output:*

When scalar is selected following options are available for the output:

1. Edge length  
Returns the length of the specific edge.
2. Face angle  
Returns the angle between the normal vectors of the faces intersecting at the edge in degree.
3. Number of connected edges  
Returns the number of edges that are connected to the given edge at both ends. In editable mesh objects the reverse edges are also counted.
4. Gaussian Curvature  
Returns the [Gaussian Curvature](#) of the mesh surface at the edge location.
5. Mean Curvature  
Returns the [Mean Curvature](#) of the mesh surface at the edge location.
6. \*Vertex indexes  
Returns the index of the two vertices in the given edge.
7. \*Face indexes  
Returns the index of the faces .For closed edges returns two indexes and for open edges returns only one item.
8. \*Index of connected edges

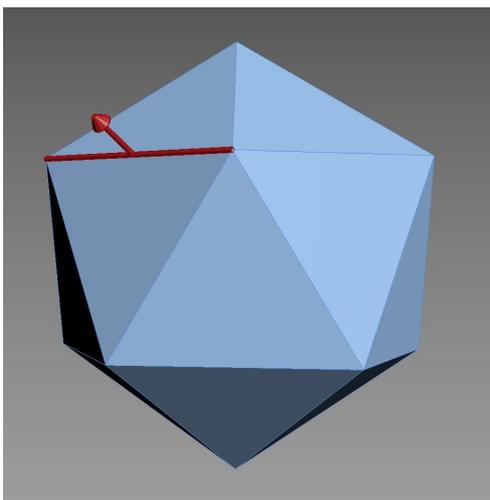
Returns the indexes of the edges that are connected to the given edge. In below example indexes of the connected edges are 4,5,7,8,22,20,19,17



*Vector Output:*

When vector is selected following options are available for the output:

1. Edge as Vector  
Returns the direction of the given edge
2. Midpoint of edge  
Returns the midpoint of the given edge. For editable meshes the result is in local coordinate system of the node.
3. Normal Vector  
The normal vector of the edge is the bisector of the angle between the faces.

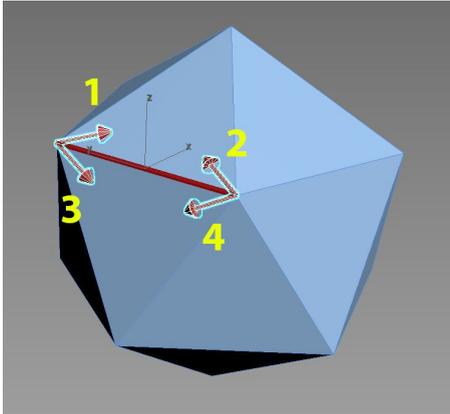


4. \*End points  
Return the position of vertices at both ends of the edge. Note that result is always contain two vectors. For Editable mesh object the position is calculated in local coordinate system of the node.
5. \*Center of adjacent faces

Controller returns the center of the adjacent faces. For open edges only one value will be returned.

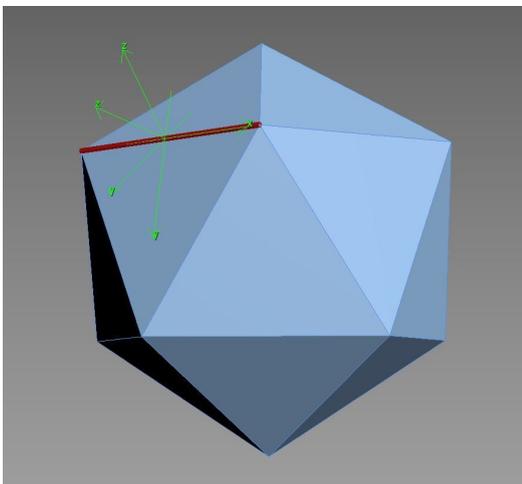
6. \*Vertex bisector

Controller returns the bisector of the angles in both end of the edge. Note in below image the order of the bisector in result array.



*Matrix outputs:*

When Matrix is selected the controller builds a transformation matrix at the center of the given edge. The X axis is parallel with edge, the Y axis parallel to the adjacent faces and the Z axis will be perpendicular to the adjacent faces.



Face

Set search area to the Face level. Note that the Face in Editable mesh objects defined by 3 edges while in Editpoly objects can have more than 3 edges.

*Scalar Output:*

When scalar is selected following options are available for the output:

1. Face area  
Returns the area of the face in system units.

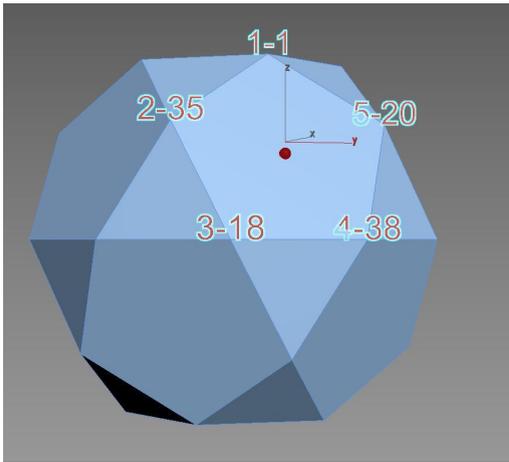
2. **Number of Edges**  
Returns the number of edges in given Face. In Editable meshes the result is always 3.
3. **Gaussian Curvature**  
Returns the [Gaussian Curvature](#) of the mesh surface at the face location.
4. **Mean Curvature**  
Returns the [Mean Curvature](#) of the mesh surface at the face location.
5. **Non-planarity**  
The non-planarity is calculated using below formula :  

$$(n - 2) * 180 - \sum_{i=1}^n \alpha$$

Where  
 $n$  is number of angles in the face  
 $\alpha$  is the interior angle in degree  
 In a planar polygon the sub of interior angles is equal to  $(n - 2) * 180$  hence the non-planarity using above formula is zero meaning the polygon is planar.
6. **Non-planarity (Quad)**  
The non-planarity claculated from below fromal is only valid for edit-poly objects with quad faces (polygons with 4 edges).  

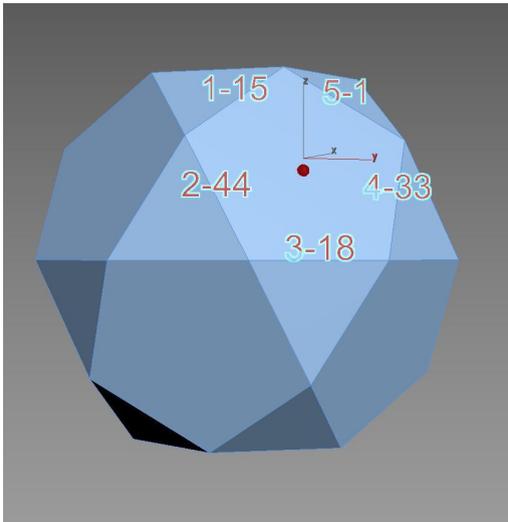
$$D/A$$

Where  
 $D$  is the shortest distance between the two diagonal of the quad face. Note if the face is planar  $D$  is zero.  
 $A$  is Face area.
7. **Total interior angle of polygon**  
Returns the sum of the interior angles in given face (poly) . if the face is planar the result is  $(n - 2) * 180$  where  $(n)$  is number of vertices.
8. **Material ID**  
Returns the material ID of the given face.
9. **\*Interior angle of vertices**  
Returns the interior angles of the given face in degree, note that result is not ordered clockwise or anti-clockwise.
10. **\*Index of vertices (ordered)**  
Returns the indexes of the vertices of a Face in order.In below example the controller returns the indexes in anti-clockwise order #(1,35,18,38,20)



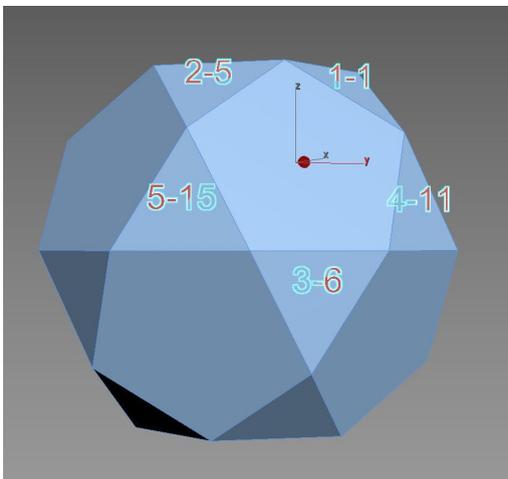
11. \*Index of edges (ordered)

Returns the indexes of the edges of a Face in order. In below example the controller returns the indexes in anti-clockwise order #(15,44,18,33,1)



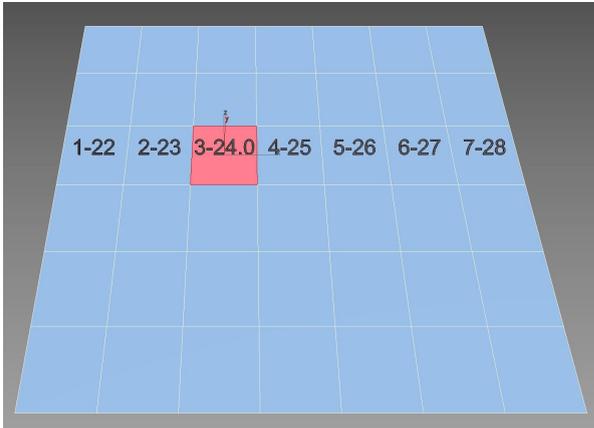
12. \*Index of adjacent faces

Returns the indexes of the adjacent faces. Not the result is not ordered. See the example below #(1,5,6,11,15)



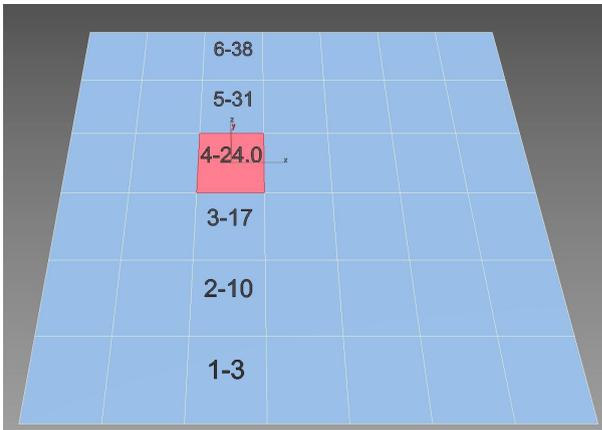
13. \*Index of Loop Faces

Returns the indexes of the faces in the same loop. The result is in order for example in below the faces in the same loop of the face 24 are #(22,23,24,25,26,27,28)



#### 14. \*Index of Ring Faces

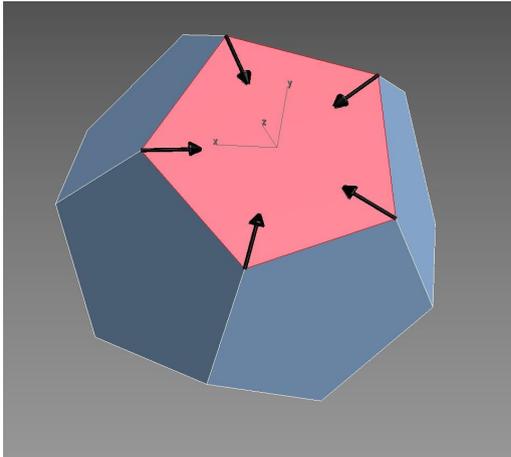
Returns the indexes of the faces in the same ring. The result is in order for example in below the faces in the same ring of the face 24 are #(3,10,17,24,31,38)



#### *Vector Output:*

When vector is selected following options are available for the output:

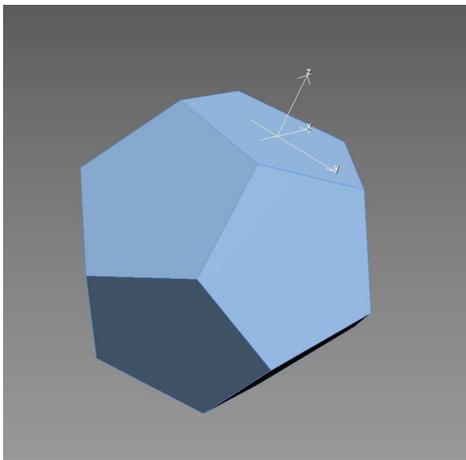
1. Face normal  
Returns the normal vector of the given face. For editable mesh objects the output is calculated in local coordinate system of the node.
2. Center point of face  
Returns the center of the face. For more information about the face center refer to MAX-Script documentation.
3. \*Actual vertex position  
Returns an array containing the position of the vertices. Note that for editable mesh objects only 3 values will be returned ( in local coordinate system of the node) . In Editable poly objects the result is always in order.
4. \*Midpoint of edges  
Returns an array containing the position of the midpoint of the edges. Note that for editable mesh objects only 3 values will be returned ( in local coordinate system of the node) . In Editable poly objects the result is always in order.
5. \*Vertex bisector  
Returns the bisector of the interior angles of the face in order.



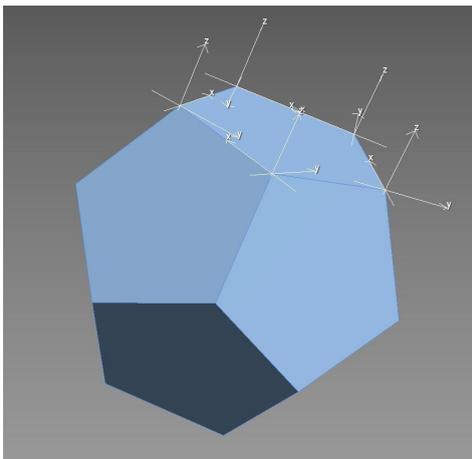
*Matrix outputs:*

When matrix is selected following options are available for the output:

1. Transformation at face center  
Builds a transformation matrix at the center of the face where the Z axis is aligned with the face normal



2. \*Transformation at face corner along the edge  
Builds a transformation matrix at the vertex location which the X axis is align with the edge and Z axis is aligned with the face normal. Note that you don't have any control on which edge is selected for X axis.



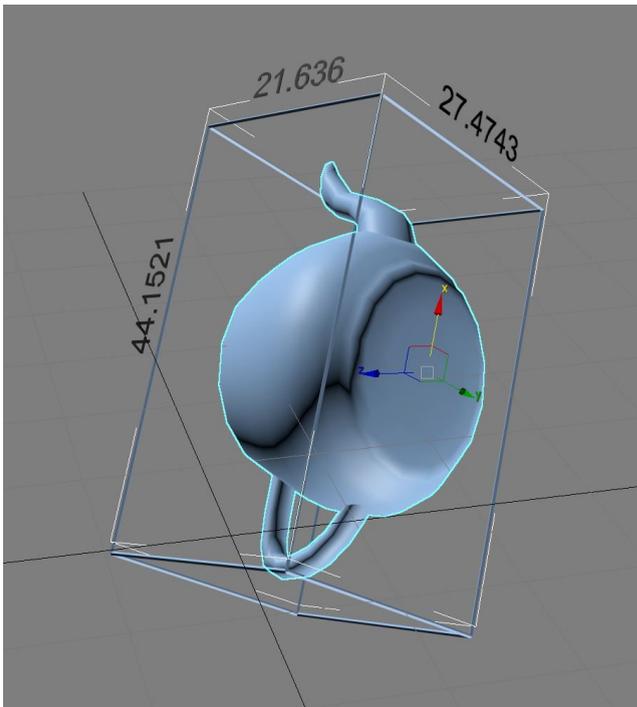
## Object

Set search area to the Object level.

### *Scalar Output:*

When scalar is selected following options are available for the output:

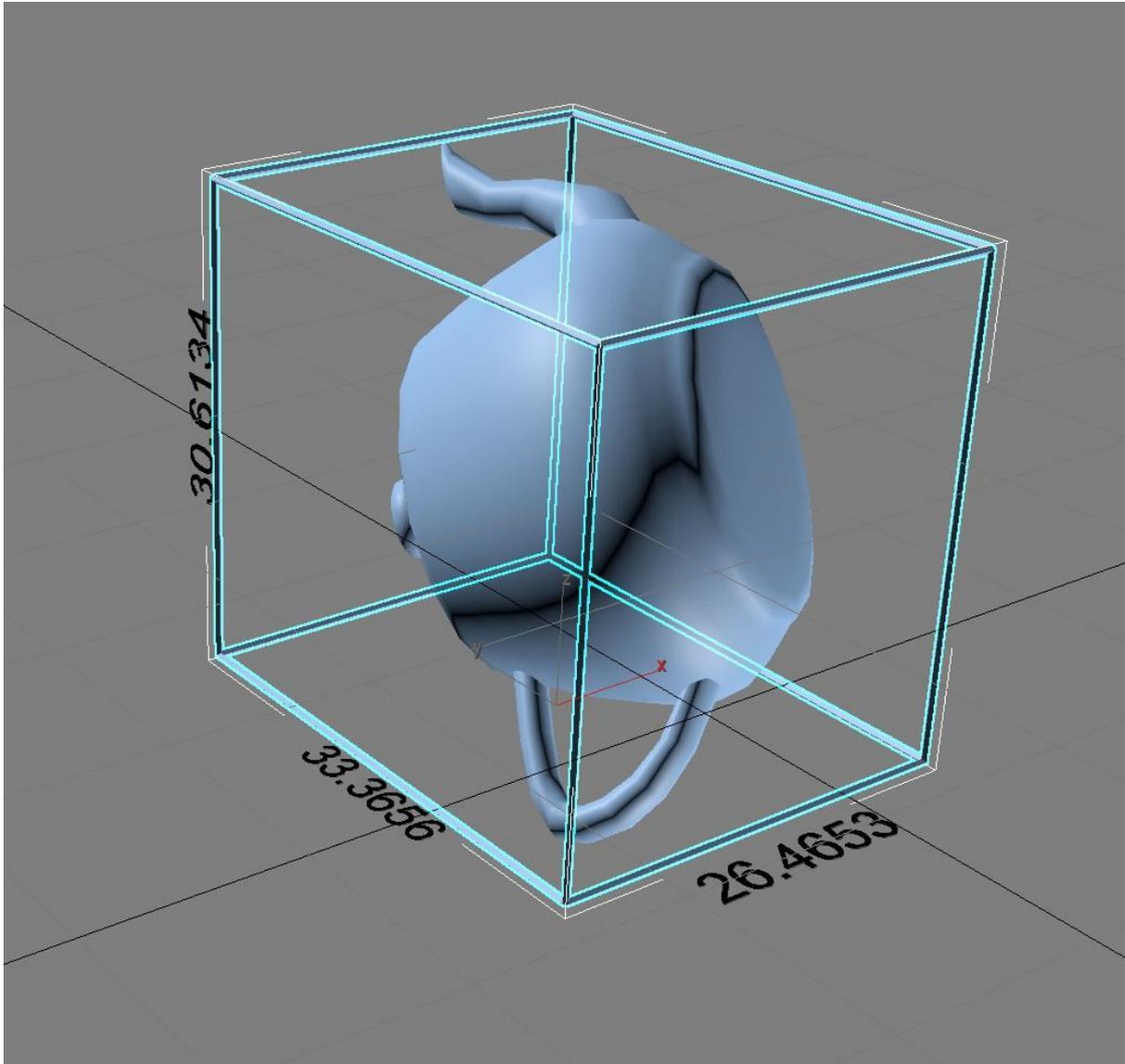
1. Total surface area  
Returns the total surface area of the object.
2. Mass volume  
Returns the mass volume of the object. The result is only valid for solid objects.
3. Total number of faces  
Returns the total number of faces. In case of Editable mesh objects the number of triangles and in the Editable Polys the number of polygons.
4. Total number of edges  
Returns the number of edges in the object. Note that in Editable meshes it includes the invisible edges plus reverse edges. See MAX-Script documentation for more information.
5. Total number of vertices  
Returns the number of vertices including the dead vertices. See more information about dead vertices in MAX-Script documentation.
6. Length (object size along local X axis)
7. Width (object size along local Y axis)
8. Height (object size along local Z axis)  
Returns the object length/Width and Height along its own local axis.  
For example in below image the above outputs are shown. The orientation of the object does not have any impact on the result.  
Length = 44.1521 , Width = 27.4743 and Height = 21.636



9. Span in X direction (object size along world X axis))
10. Span in Y direction (object size along world Y axis))

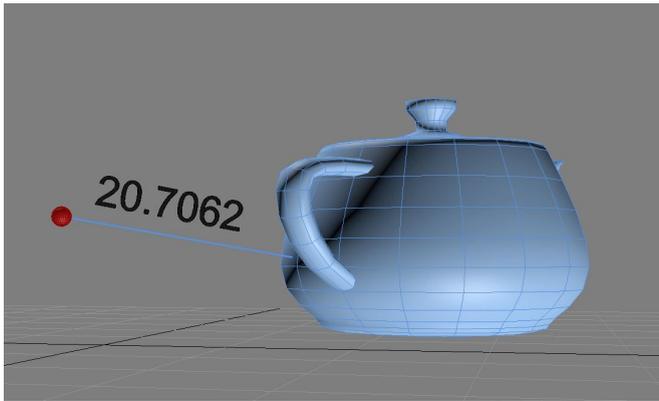
11. Span in Z direction (object size along world Z axis))

Returns the length/width and height of the global bounding box. Note that the orientation of the object affects the end result.

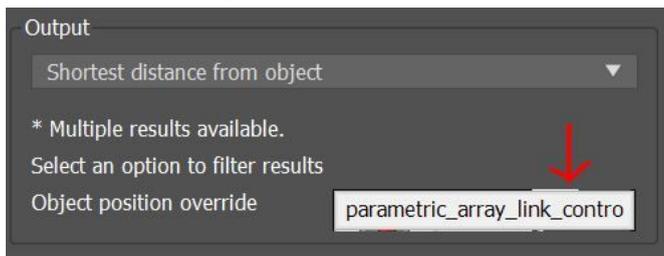


12. Shortest distance from object

Returns the shortest distance from the given point in "position override Channel" to the object.



If no controller is provided then object position in world coordinate system will be considered as the reference point.



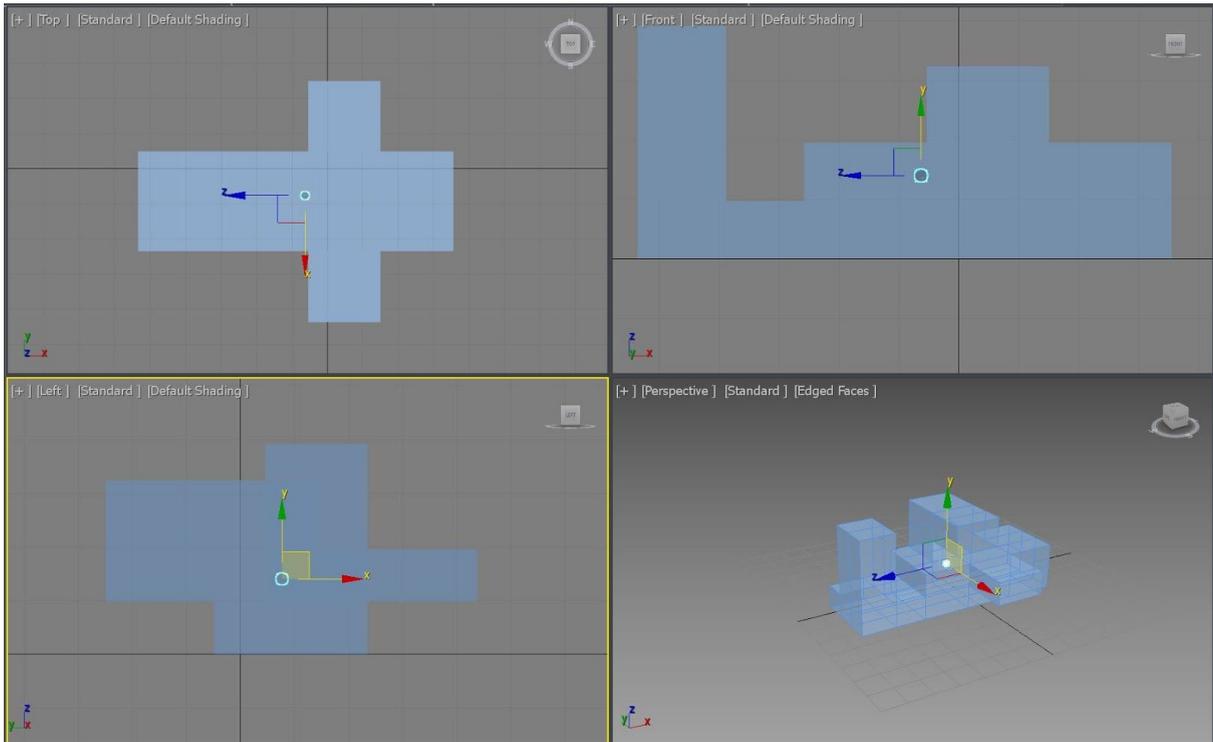
### 13. Index of closest face to the object

Return the index of the closest face in Editable mesh to the given point. The index of the face belongs to the editable Mesh object only and not the Editable poly faces. Using the index you can find the closes face. Similar to above if no controller is provided for position override channel then the object position in world coordinate system will be considered.

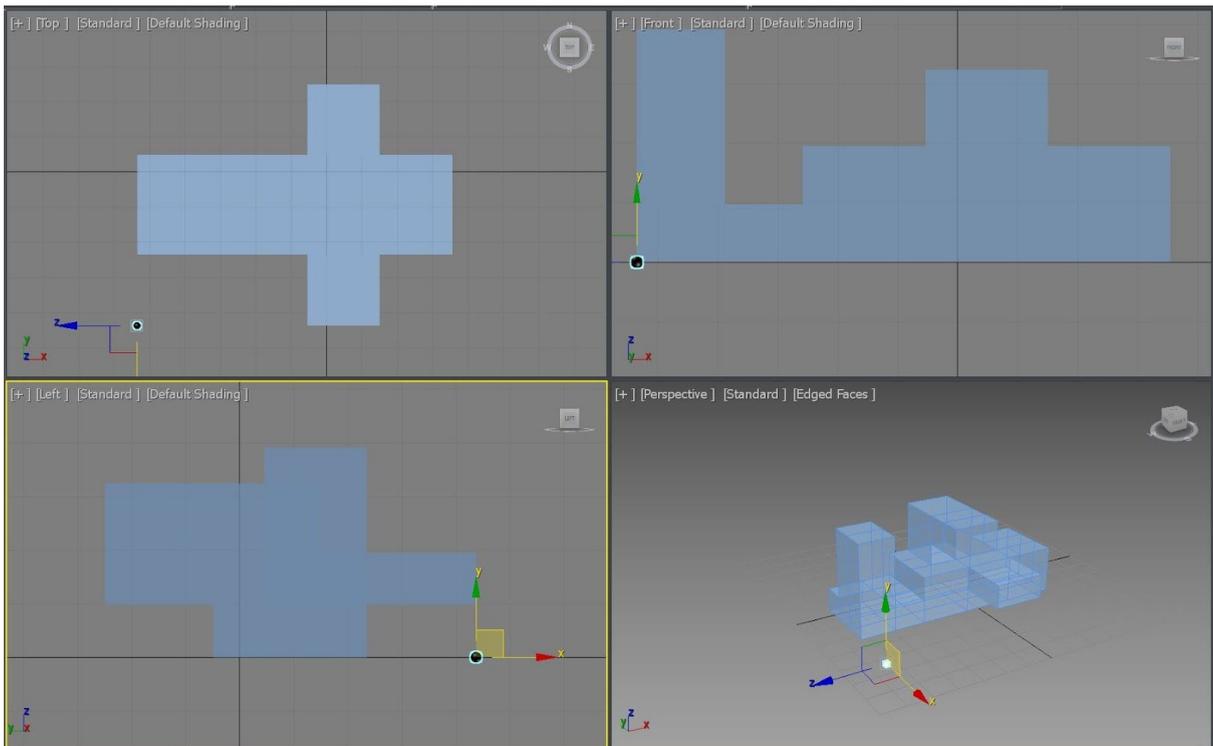
#### *Vector Output:*

When vector is selected following options are available for the output:

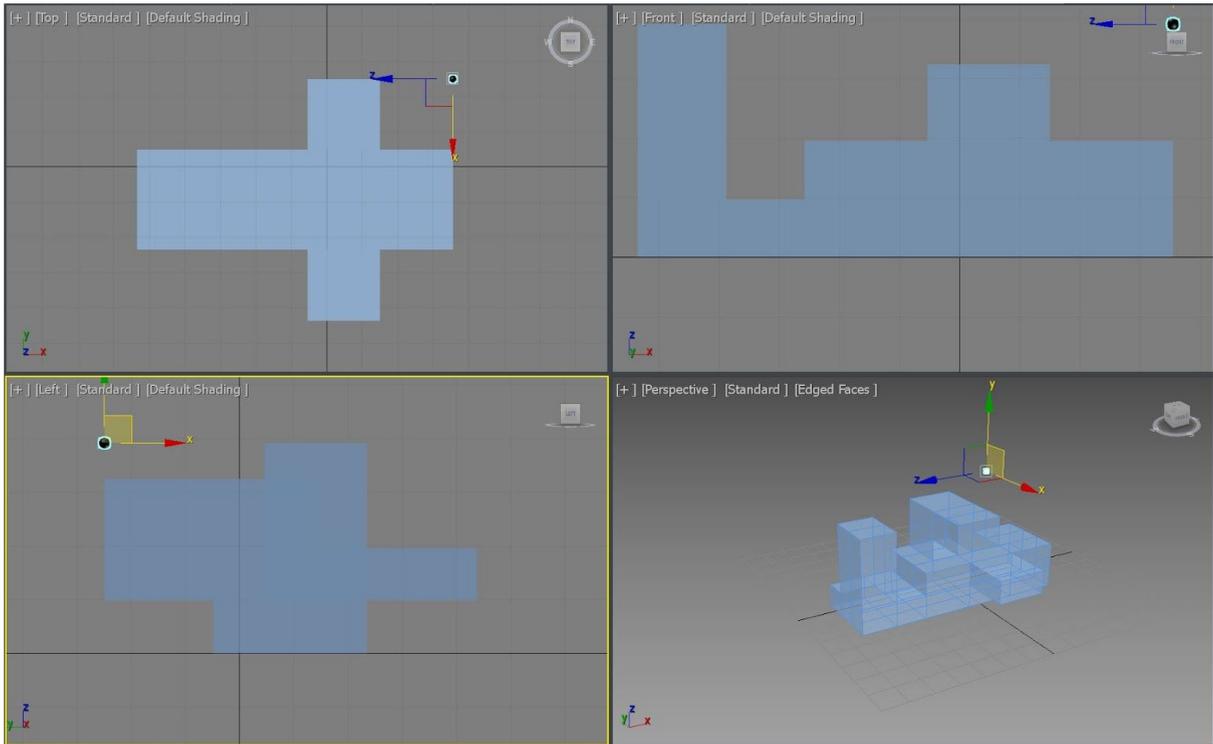
1. Center of mass



## 2. Lower corner of bounding box



## 3. Upper corner of bounding box



#### 4. Closest point to the object

Return the closest point on the Editable mesh to the given point. If no controller is provided for position override channel then the object position in world coordinate system will be considered.

#### *Matrix output:*

When matrix is selected the controller builds a transformation matrix as the center of the object and aligned with local coordinate system of the object.