AnyCAD.net

AnyCAD Part Editor User Manual

Modeling via AnyCAD script language

AnyCAD.Net
3/18/2014

Contents

1. Overview

AnyCAD Part Editor (APE) is the IDE for AnyCAD script modeling language, which can be used to create parametric 3d modes.

1.1. Modeling API

- Primitive modeling for point, line, polyline, spline, arc, circle, sphere, box, cylinder, cone, etc…
- Compound: wire, shell, solid, compound.
- Modeling methods for extrude, sweep, loft, revole, etc…
- Boolean operation for union, cut, common.

1.2. Visualization API

- You can add the material for the model and enhance the visual effect.
- APE supports dozens of embedded visualization materials.

1.3. Parameter API

- APE allows you to add parameters for the model and access the parameters in the modeling script.
- You can use the parameters to control the geometry and visualization of your models.

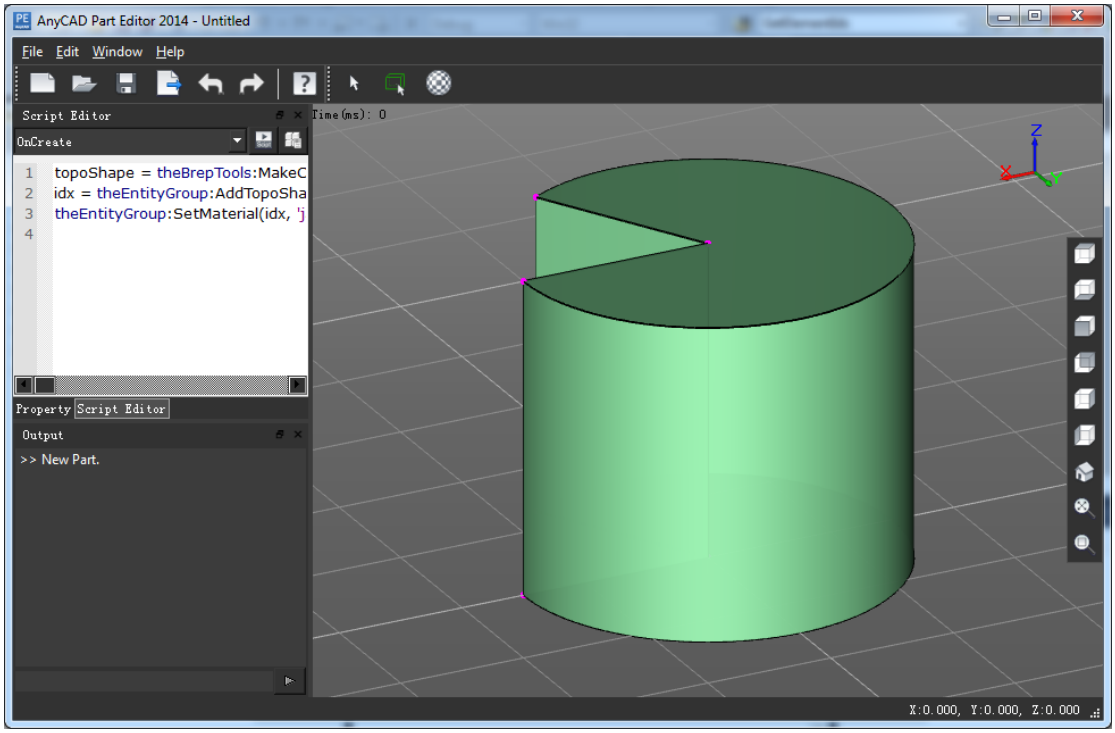
1.4. Data exchange

APE supports to export the model as the following file formats:

- STEP
- IGES
- STL

2. User Interface

AnyCAD Part Editor is consistent with the script editor view, the 3d visualization view, the output view and the property view.



3. Modeling

AnyCAD script modeling language is based on the Lua programming language, and you need to know the basic syntax of Lua. If not, that's no problem, you can learn it from the sample code of APE.

The 'Hello World' program of APE:

```
-- AnyCAD Part Script
local aTopoShape = theBrepTools:MakeBox(Vector3(0,0,0), Vector3(0,0,1), Vector3(10, 10, 20))
theEntityGroup:AddTopoShape(aTopoShape)
```

After coding, you need to press execute button  to run the script. You will see the model in the 3d view if no error happens, or the error message will show in the output view.

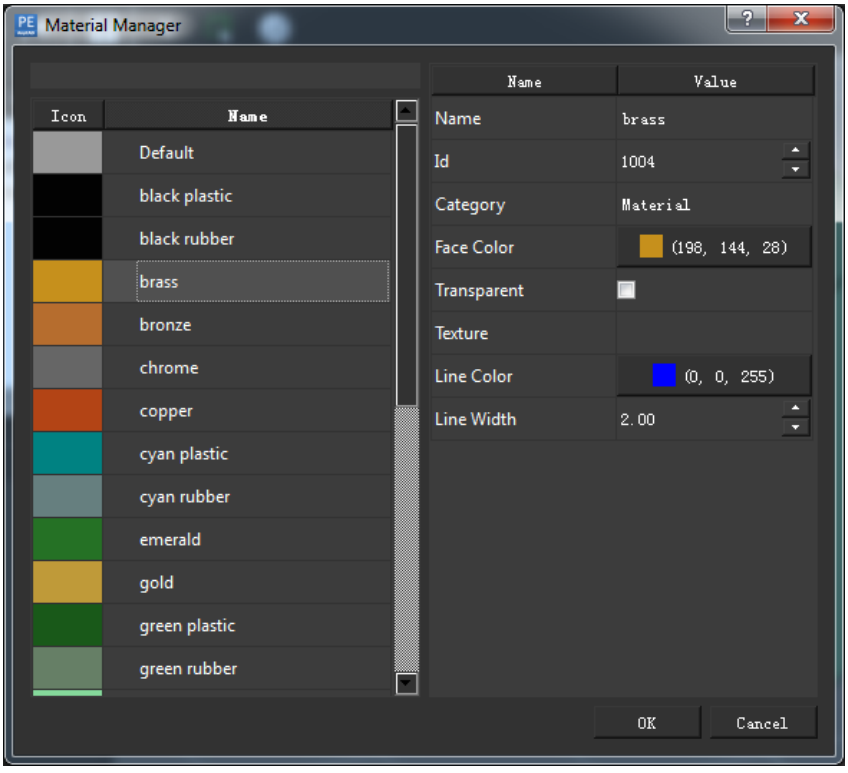
3.1. System Variables

theBrepTools : the instance of class BrepTools, which contains all the modeling method.
theEntityGroup: the instance of class EntityGroup, which is used to store the geometry.
theParameterSet: the instance of class ParameterValueSet, which is used to get/set the parameters of the model.

Please reference the API document to get the full usage of the classes.

3.2. Visual Materials

The full system visual materials can be gotten from the Material Manager dialog via button  :

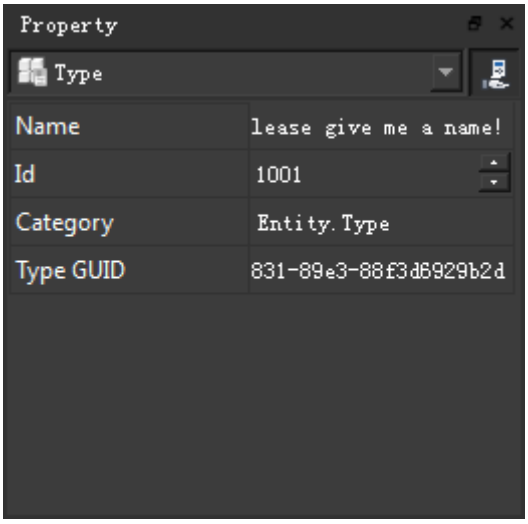


Use material in the script:

```
-- AnyCAD Part Script
local aTopoShape = theBrepTools:MakeBox(Vector3(0,0,0), Vector3(0,0,1), Vector3(10,10,10))
local idx = theEntityGroup:AddTopoShape(aTopoShape)
theEntityGroup:SetMaterial(idx,"brass ")
```

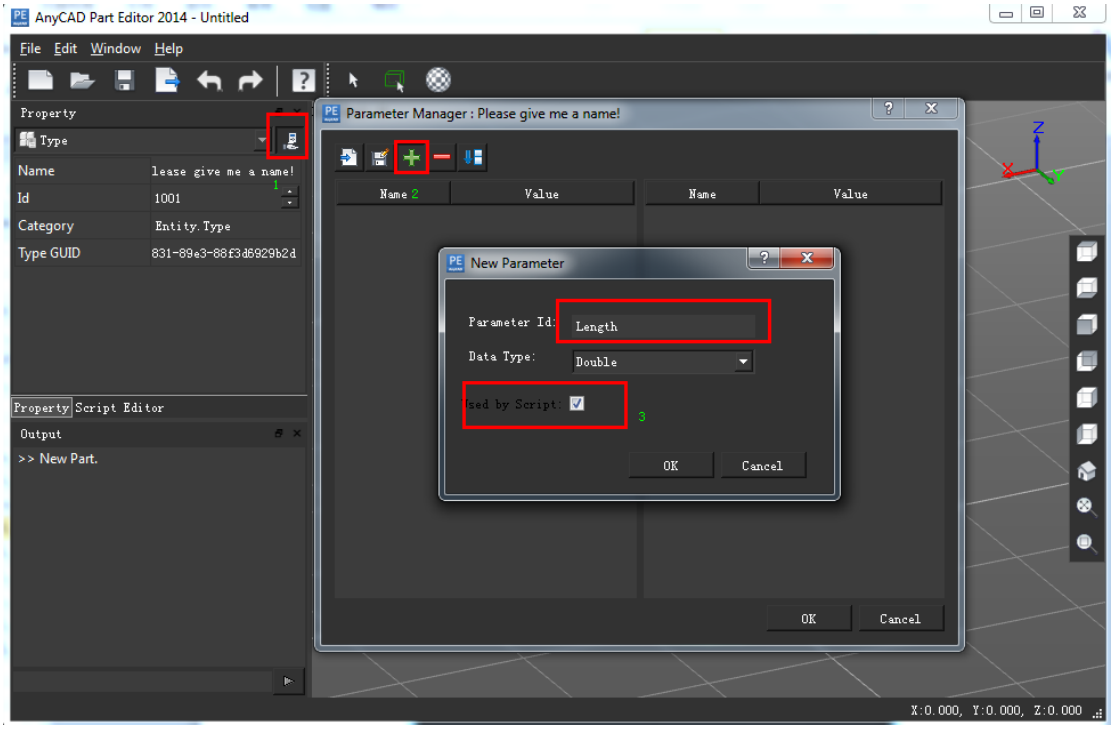
4. Model Parameters

APE supports several parameter types: string, int, double, Vector3, color…



4.1. Parameter Management

Show the parameter manage dialog and add a parameter:



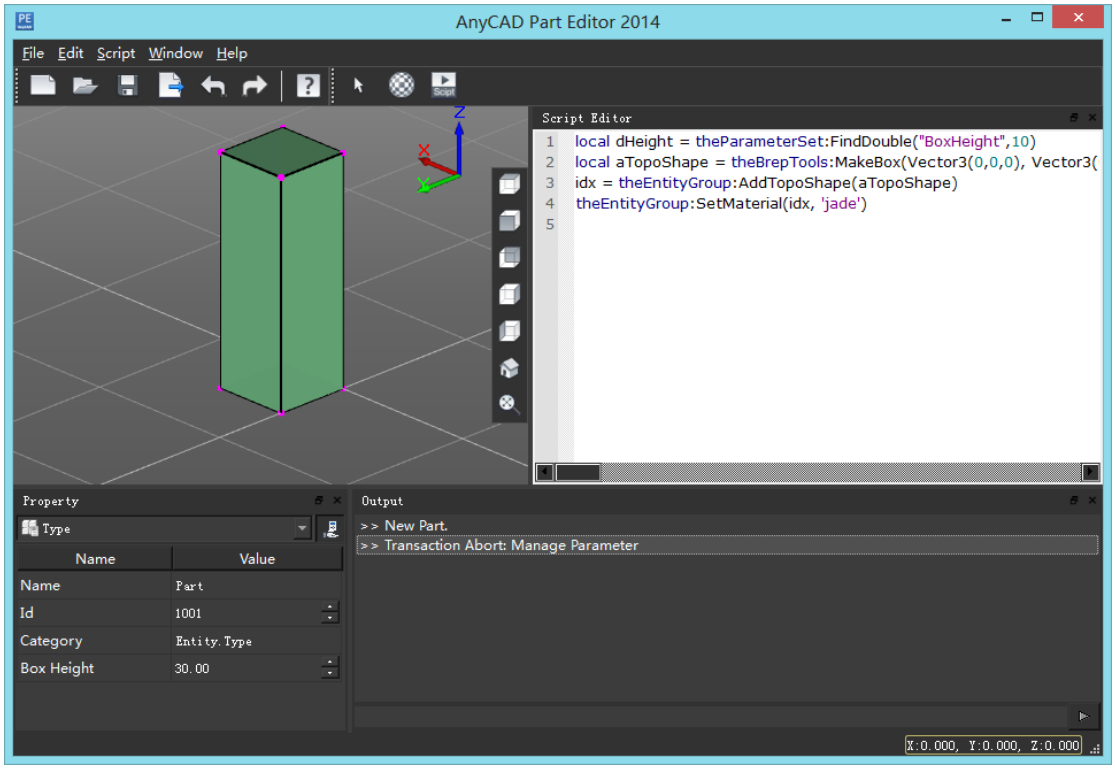
Note: if the parameter is used by the script, the option “Used by Script” should be checked.

4.2. Use the parameter in script

Suppose you have added the double parameter “BoxHeight”:

```
-- AnyCAD Part Script
local dHeight = theParameterSet:FindDouble("BoxHeight",10)
local aTopoShape = theBrepTools:MakeBox(Vector3(0,0,0), Vector3(0,0,1), Vector3(10, 10, dHeight))
theEntityGroup:AddTopoShape(aTopoShape)
```

theParameterSet provides FindDouble, FindInt, FindBool, FindString and FindVector3 methods to find the parameter value.

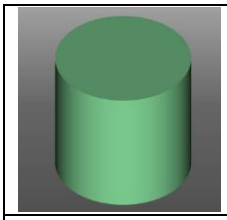


5. Data Exchange

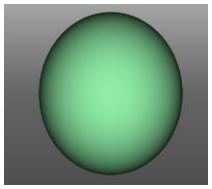
You can export the model to other file format via File->Export.

- STEP
- IGES
- STL

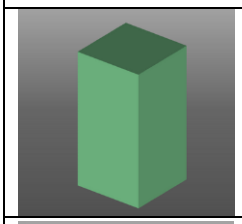
6. Sample Code



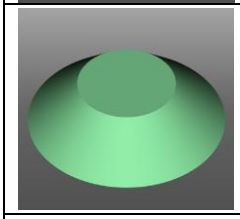
```
topoShape = theBrepTools:MakeCylinder(Vector3(0,0,0), Vector3(0,0,1), 30, 50, 0)
idx = theEntityGroup:AddTopoShape(topoShape)
```



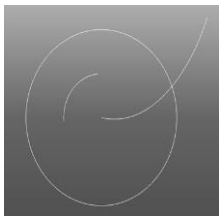
```
topoShape = theBrepTools:MakeSphere(Vector3(0,0,0), 100)
idx = theEntityGroup:AddTopoShape(topoShape)
```



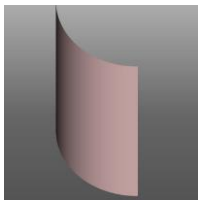
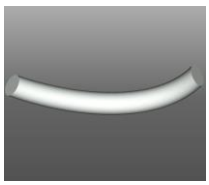
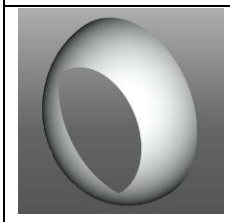
```
topoShape = theBrepTools:MakeBox(Vector3(0,0,0), Vector3(0,0,1), Vector3(50,50,80))
idx = theEntityGroup:AddTopoShape(topoShape)
```



```
topoShape = theBrepTools:MakeCone(Vector3(0,0,0), Vector3(0,0,1), 100,50, 50,0)
idx = theEntityGroup:AddTopoShape(topoShape)
```



```
-- Circle
topoShape = theBrepTools:MakeCircle(Vector3(0,0,0), 100, Vector3(0,0,1))
theEntityGroup:AddTopoShape(topoShape)
-- Arc
arc = theBrepTools:MakeArc(Vector3(-50,0,0), Vector3(0, 50,0), Vector3(0,0,0))
theEntityGroup:AddTopoShape(arc)
-- Spline
pt1 = Vector3(0,0,0)
pt2 = Vector3(50,0,0)
pt3 = Vector3(100,30,0)
pt4 = Vector3(150,100,0)
pts = Vector3List()
pts:Add(pt1)
pts:Add(pt2)
pts:Add(pt3)
pts:Add(pt4)
spline = theBrepTools:MakeSpline(pts)
theEntityGroup:AddTopoShape(spline)
```

	arc = theBrepTools:MakeArc(Vector3(-50,0,0), Vector3(0, 50,0), Vector3(0,0,0)) extrude = theBrepTools:Extrude(arc, 100, Vector3(0,0,1)) idx = theEntityGroup:AddTopoShape(extrude)
	-- Section circle = theBrepTools:MakeCircle(Vector3(0,0,0), 10, Vector3(1,0,0)) edgeGroup = TopoShapeGroup() edgeGroup:Add(circle) wire = theBrepTools:MakeWire(edgeGroup) -- Path pt1 = Vector3(0,0,0) pt2 = Vector3(50,0,0) pt3 = Vector3(100,30,0) pt4 = Vector3(150,100,0) pts = Vector3List() pts:Add(pt1) pts:Add(pt2) pts:Add(pt3) pts:Add(pt4) spline = theBrepTools:MakeSpline(pts) sweep = theBrepTools:Sweep(wire, spline) theEntityGroup:AddTopoShape(sweep)
	arc = theBrepTools:MakeArc(Vector3(-50,0,0), Vector3(0, 50,0), Vector3(0,0,0)) revol = theBrepTools:Revol(arc, Vector3(0,0,0), Vector3(1,0,1), 300) theEntityGroup:AddTopoShape(revol)

7. Core API

BrepTools	
TopoShape	MakeLine (const Vector3 &start, const Vector3 &end)

TopoShape	MakeCircle (const Vector3 &center, double radius, const Vector3 &dir)
TopoShape	MakeEllipse (const Vector3 &center, double majorRadius, double minorRadius, const Vector3 &dir)
TopoShape	MakeArc (const Vector3 &start, const Vector3 &end, const Vector3 &center)
TopoShape	MakeArc (const Vector3 &center, Real radius, Real startAngle, Real endAngle, const Vector3 &dir)
TopoShape	MakeArc3Pts (const Vector3 &start, const Vector3 &end, const Vector3 &middle)
TopoShape	MakeEllipseArc (const Vector3 &center, Real majorRadius, Real minorRadius, Real startAngle, Real endAngle, const Vector3 &dir)
TopoShape	MakeSpline (const Vector3List &pts)
TopoShape	MakePolyline (const Vector3List &pts)
TopoShape	MakePolygon (const Vector3List &pts)
TopoShape	MakePolygon (const Vector3List &pts, Real radius)
TopoShape	MakeMeshShell (const Vector3List &pts)
TopoShape	MakeWire (const TopoShapeGroup &edges)
TopoShape	MakeFace (const TopoShape &wire)
TopoShape	MakePlaneFace (const Vector3 &origion, const Vector3 &dir, Real

	minU, Real maxU, Real minV, Real maxV)
TopoShape	MakeSurfaceFromPoints (const Vector3List &arrPoints)
TopoShape	FillFace (const Vector3List &polygon)
TopoShape	MakeShell (const TopoShapeGroup &faces)
TopoShape	MakeSolid (const TopoShapeGroup &faceShells)
TopoShape	MakeCompound (const TopoShapeGroup &shapes)
TopoShape	MakeSphere (const Vector3 &center, double radius)
TopoShape	MakeBox (const Vector3 &start, const Vector3 &dir, const Vector3 &size)
TopoShape	MakeBox (const Vector3 &start, const Vector3 &end, double width, double height)
TopoShape	MakeCylinder (const Vector3 &center, const Vector3 &dir, double radius, double height, double degree)
TopoShape	MakeCone (const Vector3 &center, const Vector3 &dir, double radius, double height, double radiusTop, double degree)
TopoShape	MakeHalfSpace (const TopoShape &face, const Vector3 &refPoint)
TopoShape	BooleanAdd (const TopoShape &pShapeA, const TopoShape &pShapeB)
TopoShape	BooleanCut (const TopoShape &pShapeA, const TopoShape &pShapeB)

TopoShape	BooleanCommon (const TopoShape &pShapeA, const TopoShape &pShapeB)
TopoShape	MakeSplit (const TopoShape &pShape, const TopoShape &splitter)
TopoShape	MakeSplit (const TopoShape &pShape, const TopoShapeGroup &splitters)
TopoShape	BodySection (const TopoShape &pBody, const Vector3 &pos, const Vector3 &dir)
TopoShape	SurfaceSection (const TopoShape &pSurfA, const TopoShape &pSurfB)
TopoShape	Extrude (const TopoShape &pShape, double height, const Vector3 &dir)
TopoShape	Revol (const TopoShape &pShape, const Vector3 &position, const Vector3 &dir, double degree)
TopoShape	Fillet (const TopoShape &pShape, double radius)
TopoShape	MakeFillet (const TopoShape &shape, const IntList &edgeIdx, const FloatList &radius)
TopoShape	Chamfer (const TopoShape &pShape, double dis1, double dis2)
TopoShape	MakeChamfer (const TopoShape &shape, const IntList &edgeIdx, const FloatList &dis1, const FloatList &dis2)
TopoShape	MakeLoft (const TopoShapeGroup &listTopoShape, bool bSolid)
TopoShape	Sweep (const TopoShape &section, const TopoShape &path)

TopoShape	MakePipe (const TopoShape &section, const TopoShape &path, int mode)
TopoShape	MakeMirror (const TopoShape &pShape, const Vector3 &startPt, const Vector3 &endPt)
TopoShape	Translate (const TopoShape &pShape, const Vector3 &vec)
TopoShape	Scale (const TopoShape &pShape, const Vector3 &center, double scale)
TopoShape	Rotation (const TopoShape &pShape, const Vector3 &dir, double degree)

EntityGroup

int	AddTopoShape (const TopoShape &pTopoShape)
TopoShape	GetTopoShape (int idx) const
Entity	GetEntity (int idx) const
bool	SetMaterial (int idx, const String &materialId)

ParameterValueSet

ParameterValue	Find (const String &paramId) const
double	FindDouble (const String &paramId, double defaultValue) const
bool	SetDouble (const String &paramId, double val)
int	FindInt (const String &paramId, int defaultValue) const

bool	SetInt (const String &paramId, int val)
bool	FindBool (const String &paramId, bool defaultValue) const
bool	SetBool (const String &paramId, bool val)
String	FindString (const String &paramId, const String &defaultValue) const
bool	SetString (const String &paramId, const String &val)
Vector3	FindVector3 (const String &paramId, const Vector3 &defaultValue) const
bool	SetVector3 (const String &paramId, const Vector3 &val)
ColorValue	FindColor (const String &paramId, const ColorValue &defaultValue) const
bool	SetColor (const String &paramId, const ColorValue &val)

8. Reference

[1] www.anycad.net
[2] http://www.anycad.net/help/AGP2014/