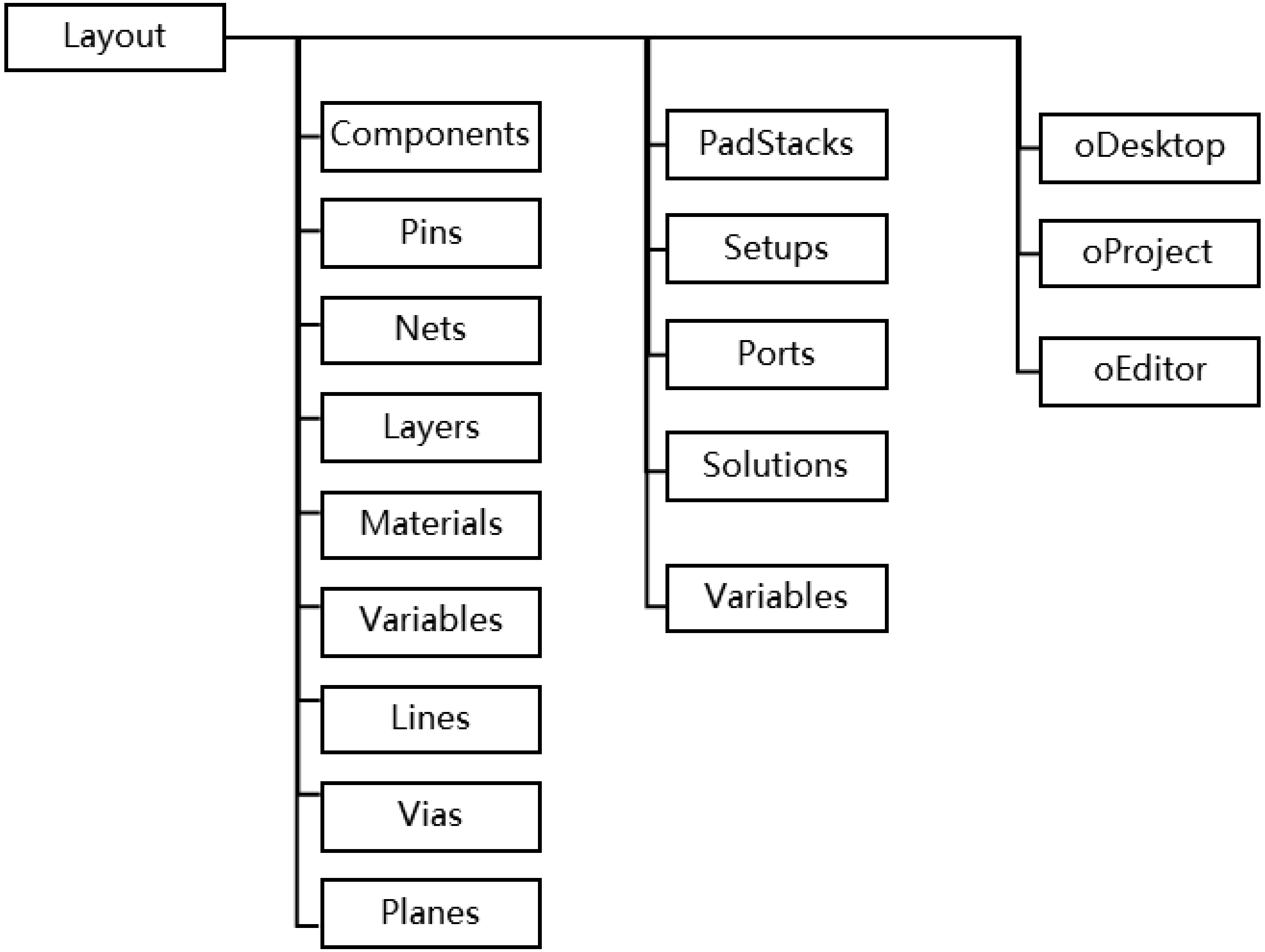
Layout对象初始化：

Layout对象代表了一块PCB Or Package设计，如果一个Project下有多个Design,可用多个Layout对象描述。



```
from pyLayout import Layout
layout = Layout("2022.2")
layout.initDesign()
```

> 1. layout = Layout()不指定版本是，pyLayout会尝试启动最新版本的AEDT界面。
> 2. 如果对应版本的AEDT是打开状态，默认使用当前AEDT窗口，不会打开新的AEDT界面。
> 3. 通过AEDT调用时(Tools->Run Script), 会默认继承当前窗口的版本号，指定的版本号不起作用。
> 4. layout.initDesign() 用于初始化Layout对象，初始化之后Layout对象的属性才有效。 Layout打开工程文件

在python环境执行时，需要按照pyAedt. (本测试环境为python环境)

In [ ]:

```python
from pyLayout import Layout
layout = Layout("2022.2")
#layout.openAedt(r"C:\work\Project\AE\Script\test_pcb\galileo.aedt")
layout.initDesign()
```

Layout对象的访问（Components）

Layout由Component, Layer, Material, Net, Pin, Line, Via, Plane, Setup, Solution, Variable等元素组成，这些元素可以通过Layout的集合对象来访问。比如Layout.Components对象代表了Layout所有Component的集合，可以用于获取所有的Component对象。
Component, Layer, Material, Net, Pin, Line, Via, Plane, Setup, Solution, Variable的集合都支持类似调用方式。 以上对象的访问不区分大小写。

方法1：使用位号索引，访问Layout上的U1器件对象

In [ ]:

```python
layout.Components["U1"]
```

方法2：直接作为属性，访问Layout上的U1器件对象

In [ ]:

```python
layout.Components.U1
```

方法3: 直接访问Layout的U1对象，

Layout会尝试U1的类型，返回Compoent对象或者其它对象，如果遍历未发下U1元素，则抛出异常。 （存在重名问题,不推荐）

In [ ]:

```
layout["U1"]
layout.U1
```

除了以上方法，可以直接使用index对器件进行访问：

In [ ]:

```
layout.Components[0]
layout.Components[0:5]
```

使用for循环迭代对象

In [ ]:

```
for comp in layout.Components:
    print(comp.Name)
```

访问Component的属性

In [ ]:

```
U1 = layout.Components["U1"]
dir(U1)
```

访问Component的pins

In [ ]:

```
for pin in U1.pins:
    print(pin.name,pin.net)
```

Layout.Layers 对象访问

获取具体某层的layer对象

layer对象可以通过层名索引，或者通过位置获取。

In [ ]:

```
layout.layers["Top"] #使用名字获取
layout.layers["C1"] #获取第一个金属层，即Top层
layout.layers["CB1"] #反向获取第一个金属层，即BOTTOM层
layout.layers["D1"] #获取第一个介质层
layout.layers["DB1"] #反向第一个介质层
layout.layers["S1"] #获取叠层所有层的第一层
layout.layers["SB1"] #反向获取叠层所有层的第一层

layout.layers.Top #使用名字获取
layout.layers.C1 #使用名字获取
layout.layers.S1 #使用名字获取
```

获取和设置Layer的属性

In [ ]:

```
layout.layers["Top"]["Thickness"] #获取层厚度
layout.layers["Top"].Thickness #获取层厚度

layout.layers["Top"].Thickness = "1.9mil" #设置层厚度

layout.layers["Top"].Thickness = "1.9mil" #设置层厚度

layout.layers["Top"].Material
layout.layers["Top"].FillMaterial
layout.layers["Top"].Height
layout.layers["Top"].Lower
```

获取和设定粗糙度

In [ ]:

```
layout.layers["Top"].Roughness = "0.5um"
```

In [ ]:

```
layout.layers["Top"].UseRoughness = True
layout.layers["Top"].Roughness = "0.5um"
layout.layers["Top"].Roughness
```

Variable 变量的增加和赋值

In [ ]:

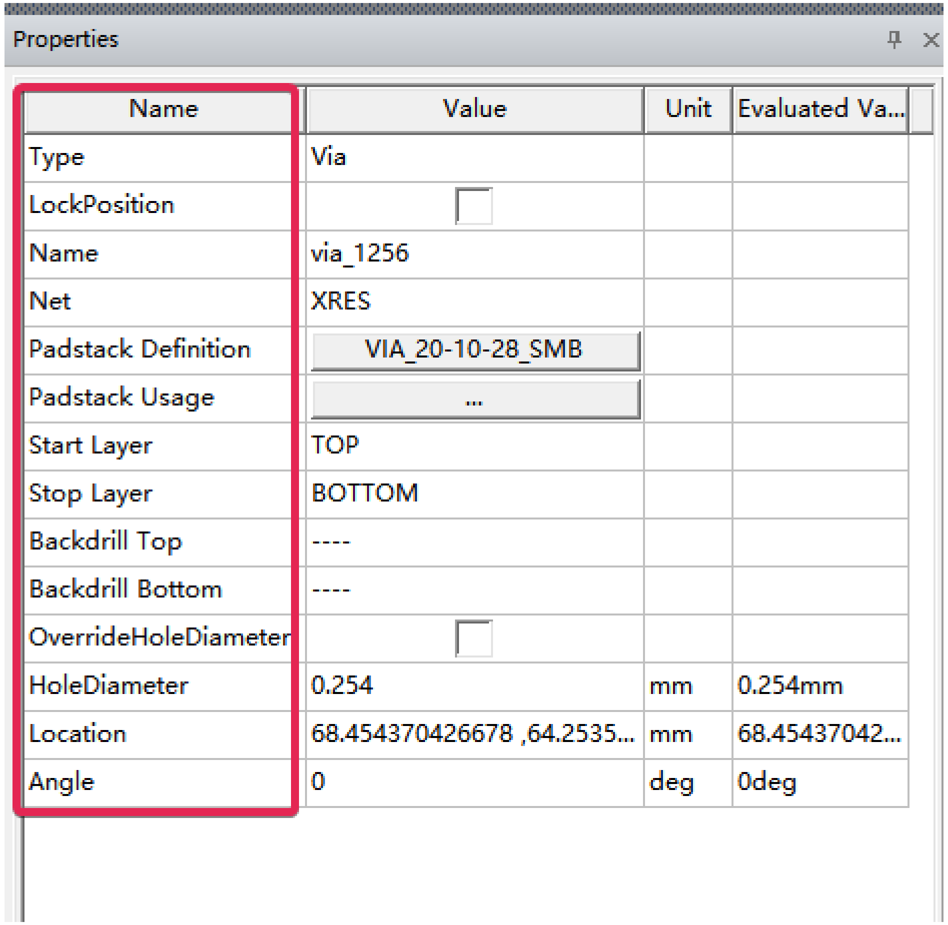
```
layout.Variables.add("test1") #局部变量
layout.Variables.add("$test2") #全局变量

layout.Variables.test1 = "10mil"
layout.Variables["$test2"] = "20mil"

print(layout.Variables["test1"])
```

In [ ]:

```
varss = layout.Variables
print(varss.All)
```

访问和管理line,via,plane,pin,net的属性

可以通道名字直接索引对象，在3D Layout UI中看到的物体属性，均可直接访问，不区分大小写。



Properties

Name	Value	Unit	Evaluated Va...
Type	Via		
LockPosition	□		
Name	via_1256		
Net	XRES		
Padstack Definition	VIA_20-10-28_SMB		
Padstack Usage	...		
Start Layer	TOP		
Stop Layer	BOTTOM		
Backdrill Top	----		
Backdrill Bottom	----		
OverrideHoleDiameter	□		
HoleDiameter	0.254	mm	0.254mm
Location	68.454370426678 ,64.2535...	mm	68.45437042...
Angle	0	deg	0deg

In [ ]:

```
via1 = layout.vias["via_1256"] #获取via对象
via2 = layout.vias.via_1256  #和via1为同一对象
print(via1 is via2)
```

对象属性的访问

可以通过key值访问，也可以作为属性访问，如果属性值有空格，允许去掉空格进行索引。不区分大小写。

In [ ]:

```
print(via1.name,via2.Net)
print(via1["Start Layer"]) #直接访问属性
print(via1["StartLayer"]) #属性可以去掉空格
print(via1.StartLayer) #和前面两种方法等同
```

line,via,plane,pin,net 等UI可见属性同Via案例

In [ ]:

```
line1 = layout.lines["line_4200"]
pin1 = layout.pins["J2L1-48"]
pin1_1 = layout.pins.J2L1_48  # 属性中的-可以转换为_
p1 = layout.planes["poly_354"]
net1 = layout.nets["M_CAS_N"]
```

setup的管理

setup 添加，获取，删除setup

In [ ]:

```
layout.setups.add("hfss1",solutionType = "HFSS")
layout.setups.add("siwave1",solutionType = "SIwave")
```

hfss setup属性访问和设定

In [ ]:

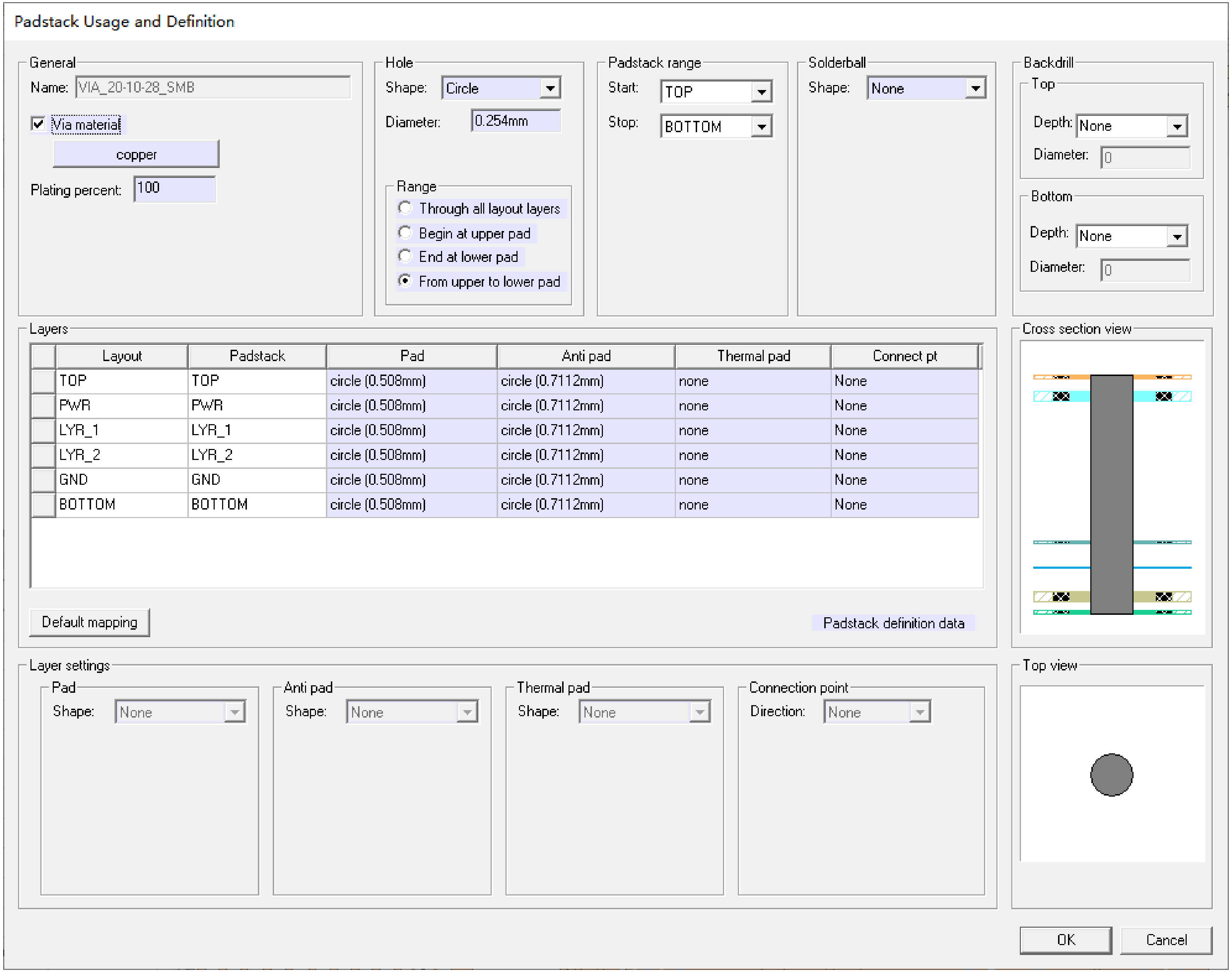
```
# dir(layout.setups["hfss1"])

layout.setups["hfss1"].AdaptiveFrequency = "10Ghz"
layout.setups["hfss1"].DeltaS = "0.01"
layout.setups["hfss1"].MaxPasses = 20

print(layout.setups["hfss1"].AdaptiveFrequency)
print(layout.setups["hfss1"].DeltaS)
print(layout.setups["hfss1"].Order)
print(layout.setups["hfss1"].MaxPasses)
```

sweep的添加和删除

In [ ]:

```
#添加HFSS Sweep
layout.setups["hfss1"].addSweep("swp1")
layout.setups["hfss1"].Sweeps["swp1"].SweepData = "LIN 0GHz 20GHz 0.01GHz"
layout.setups["hfss1"].Sweeps["swp1"].UseQ3D = True
layout.setups["hfss1"].Sweeps["swp1"].InterpolatingTolerance = 0.001 #0.1%
layout.setups["hfss1"].Sweeps["swp1"].SweepType = "interpolating" #default


#添加SIwave Sweep
layout.setups["siwave1"].addSweep("swp1")
layout.setups["siwave1"].Sweeps["swp1"].SweepData = "LIN 0GHz 20GHz 0.01GHz"
layout.setups["siwave1"].Sweeps["swp1"].UseQ3D = True
layout.setups["siwave1"].Sweeps["swp1"].InterpolatingTolerance = 0.001 #0.1%
layout.setups["siwave1"].Sweeps["swp1"].SweepType = "interpolating" #default
```

PadStack的访问



Padstack Usage and Definition

General

Name: VIA_20-10-28_SMB

☑ Via material

copper

Plating percent: 100

Hole

Shape: Circle

Diameter: 0.254mm

Range

○ Through all layout layers

○ Begin at upper pad

○ End at lower pad

● From upper to lower pad

Padstack range

Start: TOP

Stop: BOTTOM

Solderball

Shape: None

Backdrill

Top

Depth: None

Diameter: 0

Bottom

Depth: None

Diameter: 0

Layers

Layout	Padstack	Pad	Anti pad	Thermal pad	Connect pt
TOP	TOP	circle (0.508mm)	circle (0.7112mm)	none	None
PWR	PWR	circle (0.508mm)	circle (0.7112mm)	none	None
LYR_1	LYR_1	circle (0.508mm)	circle (0.7112mm)	none	None
LYR_2	LYR_2	circle (0.508mm)	circle (0.7112mm)	none	None
GND	GND	circle (0.508mm)	circle (0.7112mm)	none	None
BOTTOM	BOTTOM	circle (0.508mm)	circle (0.7112mm)	none	None

Default mapping

Padstack definition data

Cross section view

Layer settings

Pad

Shape: None

Anti pad

Shape: None

Thermal pad

Shape: None

Connection point

Direction: None

Top view

OK

Cancel

In [ ]:

```
padStk1_name = layout.Vias.via_2105["Padstack Definition"]
padStk1 = layout.PadStacks[padStk1_name]
print(padStk1.DrillSize)
print(padStk1["Top"].PadSize)
print(padStk1["Top"].AntipadPadSize)
print(padStk1["Top"].ThermalPadSize)

# LayerName Top可以使用C1进行索引，即Concudtor的第一层
print(padStk1["C1"].PadSize)
print(padStk1["C1"].AntipadPadSize)
print(padStk1["C1"].ThermalPadSize)
```

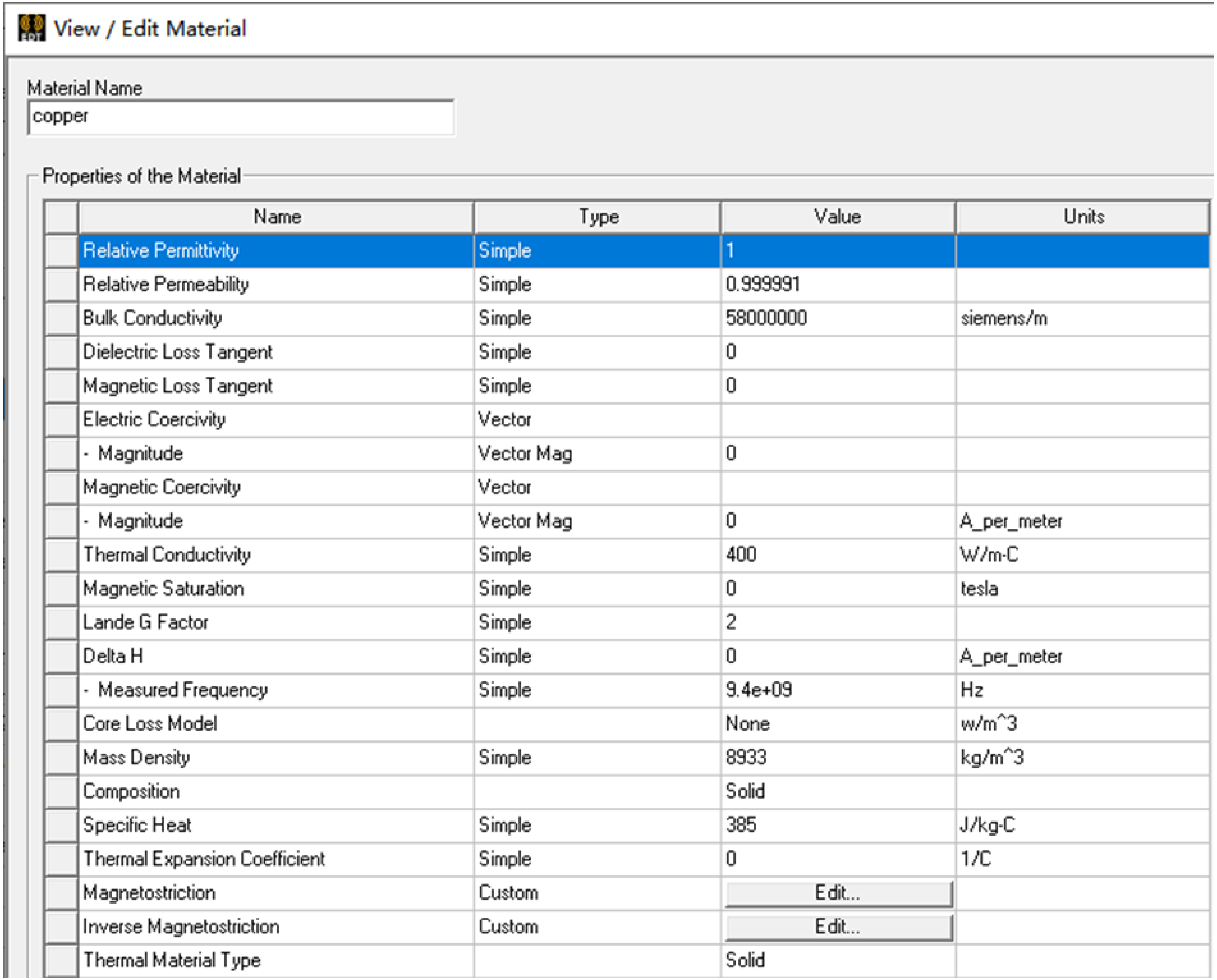
按照属性直接访问

In [ ]:

```
print(padStk1["Top"].pad.shp)
print(padStk1["Top"].pad.Szs)

print(padStk1["Top"].ant.shp)
print(padStk1["Top"].ant.Szs)

print(padStk1["Top"].thm.shp)
print(padStk1["Top"].thm.Szs)
```

Material属性访问

可以按照材料的属性名作为key值进行索引



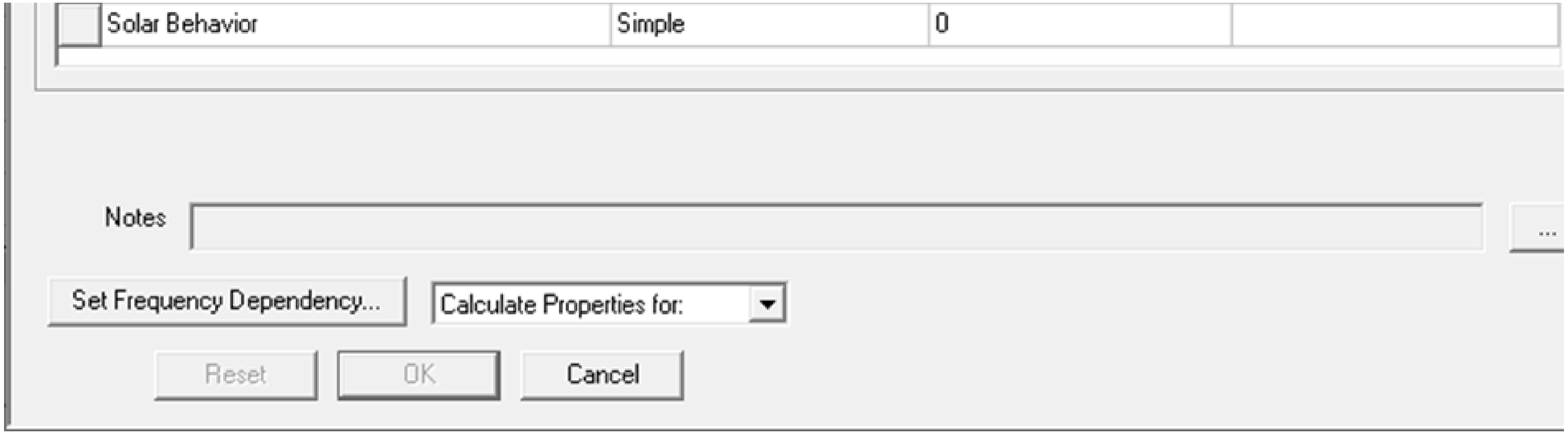
View / Edit Material

Material Name

copper

Properties of the Material

Name	Type	Value	Units
Relative Permittivity	Simple	1	
Relative Permeability	Simple	0.999991	
Bulk Conductivity	Simple	58000000	siemens/m
Dielectric Loss Tangent	Simple	0	
Magnetic Loss Tangent	Simple	0	
Electric Coercivity	Vector		
· Magnitude	Vector Mag	0	
Magnetic Coercivity	Vector		
· Magnitude	Vector Mag	0	A_per_meter
Thermal Conductivity	Simple	400	W/m-C
Magnetic Saturation	Simple	0	tesla
Lande G Factor	Simple	2	
Delta H	Simple	0	A_per_meter
· Measured Frequency	Simple	9.4e+09	Hz
Core Loss Model		None	w/m^3
Mass Density	Simple	8933	kg/m^3
Composition		Solid	
Specific Heat	Simple	385	J/kg-C
Thermal Expansion Coefficient	Simple	0	1/C
Magnetostriction	Custom	Edit...	
Inverse Magnetostriction	Custom	Edit...	
Thermal Material Type		Solid	



Solar Behavior	Simple	0	

Notes

```
In [ ]: mat1 = layout.Materials["copper"]
        dir(mat1)
```

```
In [ ]: print(mat1["permittivity"])
        print(mat1.DK) #同 permittivity

        print(mat1["dielectric_loss_tangent"])
        print(mat1.DF) #同 dielectric_loss_tangent

        print(mat1["conductivity"])
        print(mat1.Cond) #同 conductivity

        print(mat1["Resistivity"]) # 1/conductivity
        print(mat1.Resistivity) # 1/conductivity

        print(mat1["permeability"])
        print(mat1.ur) #同 permeability
```