AnsysToolbox Help

None

Yongsheng.guo@ansys.com

Yongsheng.guo@ansys.com

Table of contents

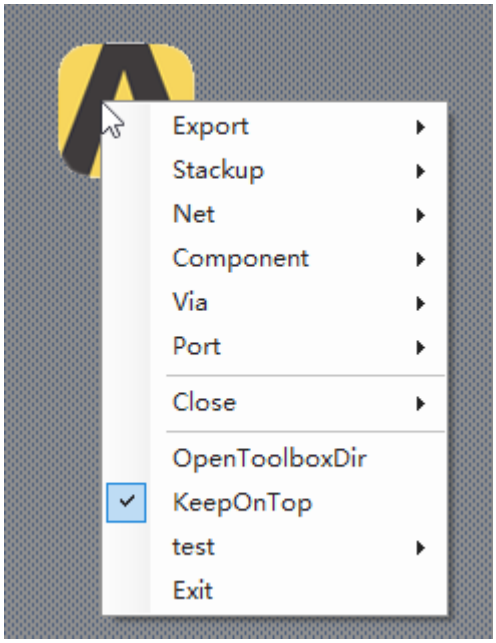
Yongsheng.guo@ansys.com

Yongsheng.guo@ansys.com

1. Welcome to Ansys Toolbox

1.1 工具说明

Ansys Toolbox提供了一种便捷的方式，让用户在AEDT中迅速运行脚本或外部程序。它通过悬浮图标和自定义菜单的组合，实现高效操作。用户可以借助XML文件快速定制菜单内容，一旦更新，这些变化将实时反映在右键菜单中，确保用户始终拥有个性化的操作体验。 整体实现效果如下：

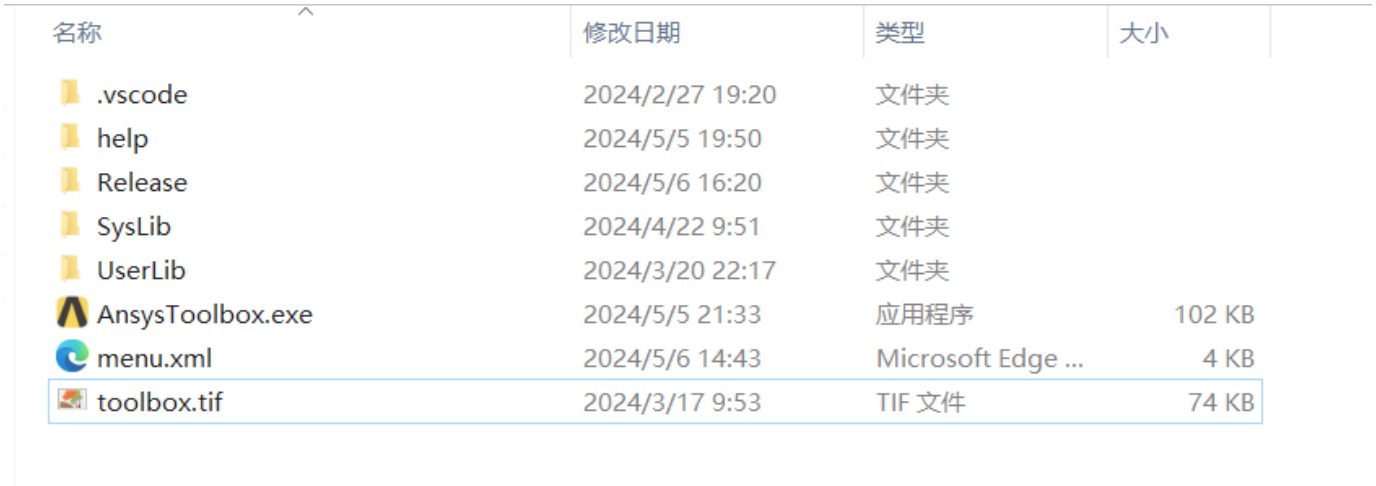


当对应菜单被点击时，Toolbox会将脚本发送至最近一次打开的AEDT窗口并执行其内容。由于Toolbox不区分AEDT的版本，因此可以兼容不同版本的AEDT，使得同一个脚本可以在不同版本的AEDT中执行。

Toolbox支持外部程序和脚本的执行，包括Exe、Python和Ironpython三种类型。目前Toolbox仅支持Windows系统。

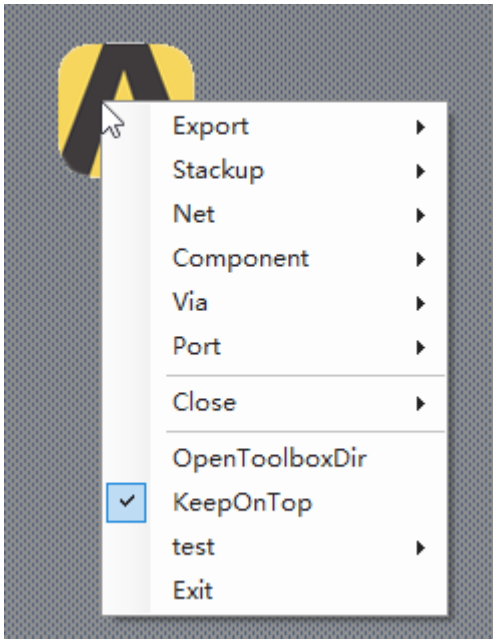
1.2 工具启动

通过目录下的"AnsysToolbox.exe"启动Toolbox，启动后Toolbox以悬浮窗口的形式显示，默认显示在所有窗体的最前面。

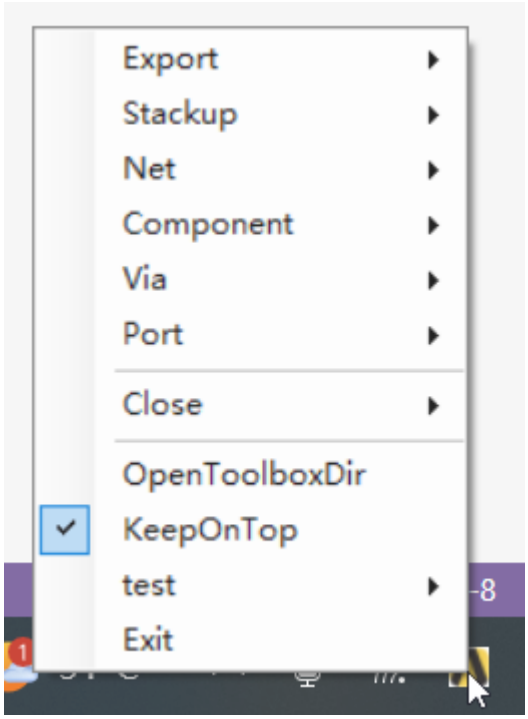


Yongsheng.guo@ansys.com

可以通过两种方式弹出右键菜单： 1. 通过在悬浮窗口点击右键



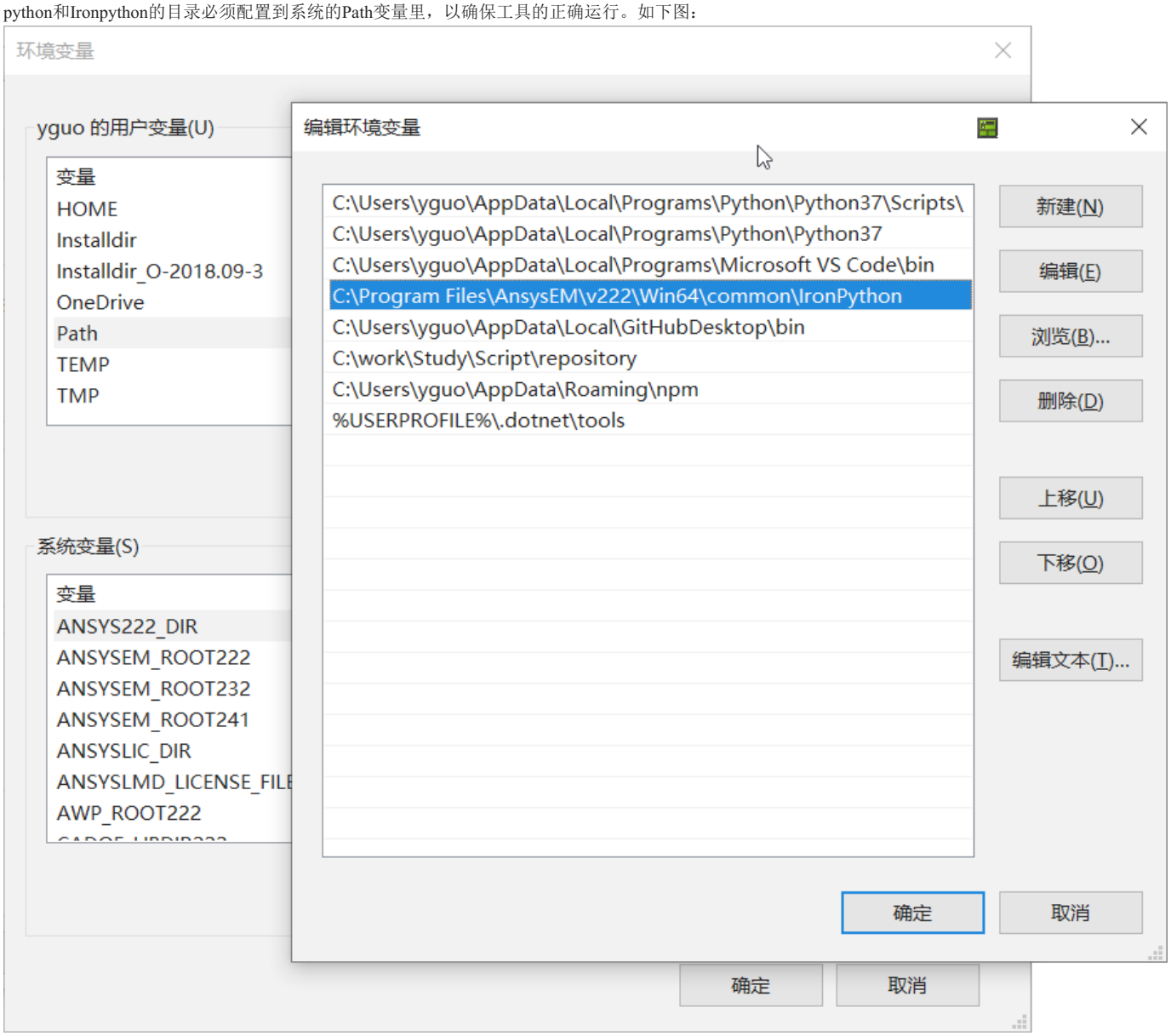
2. 通过通知图标点击右键。



Yongsheng.guo@ansys.com

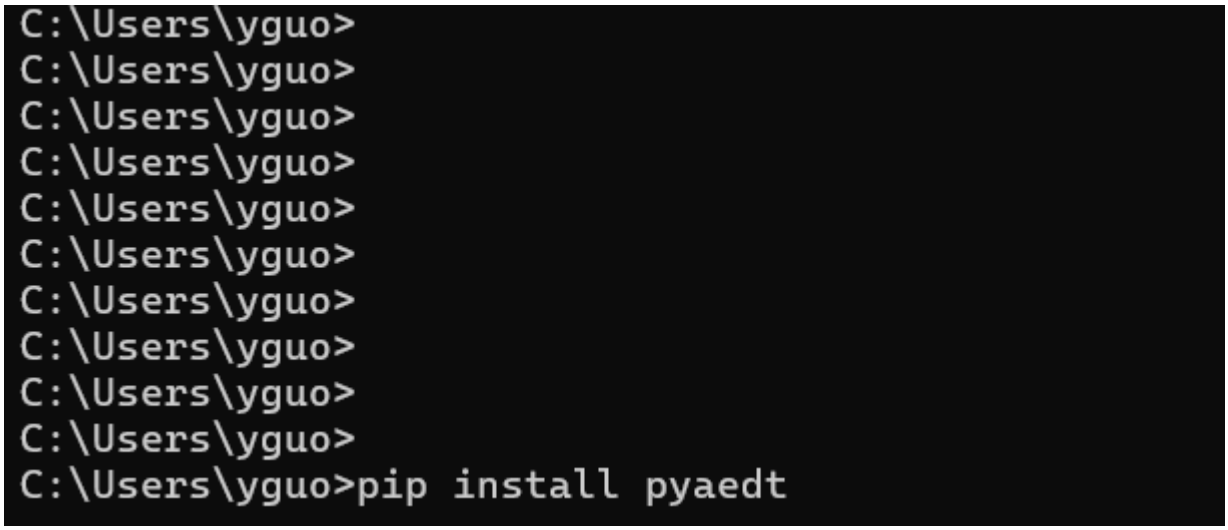
1.3 Python环境配置

python和Ironpython的目录必须配置到系统的Path变量里，以确保工具的正确运行。如下图：



对于python脚本的执行，pyaedt必须按照，可以通过在命令窗口中输入：`pip install pyaedt`

Yongsheng.guo@ansys.com



1.4 用户子定义菜单和脚本

用户可以根据自身的需要添加菜单和脚本，定制符合本公司需求的Toolbox工具集合。可以参照下面的方法对菜单和脚本内容继续增加和删除。

1.5 菜单的设定更新

用户菜单可以通过目录下的menu.xml进行更新和设定。Toolbox发布时已经附带了一些功能和菜单项，用户可以根据需要进行删减和添加自己的菜单内容。

菜单格式如下：

```
<Menu>
    <SubMenu Name="test">
        <SubMenu Type="MenuItem" Name="showProjectName_python" ExecuteType="Python" Path="$UserLib/Template/showProjectName.py" Arguments ="" PythonPath=""></
SubMenu>
        <SubMenu Type="MenuItem" Name="showProjectName_Ironpython" ExecuteType="IronPython" Path="$UserLib/Template/showProjectName.py" Arguments =""
EntryFunc="main" PythonPath=""></SubMenu>
    </SubMenu>
</Menu>
```

1.5.1 参数说明

- 时对菜单项的声明，允许进行嵌套，嵌套将以子菜单的形式呈现。子菜单的深度未作限定，但是不建议嵌套的过深，影响用户体验。
- Type：取值可以为 ″MenuItem"（菜单项）,"Separator"(分隔符), 省略时默认为″MenuItem"
- Name：菜单显示的名称
- ExecuteType：可执行程序的类型，可选值：Python，IronPython，EXE。 Command类型为内部保留类型，用户无法进行定义。
- Path: 可执行程序，脚本的路径。可以使用绝对路径值。如果使用相对路径，可以使用″$+目录″的形式引入当面目录下的文件夹。
- Arguments: 允许传递参数给可执行脚本，多个参数以空格隔开。
- PythonPath： 可以指定Python的执行路径，特别是存在多个版本时可以按照路径区分版本。省略时会从Path变量中查找Python执行文件。
- EntryFunc： 针对Python，IronPython指定运行脚本的入口函数，默认为″main"函数（可以省略），如果为其它函数则需要指定函数名称。

1.5.2 注意事项

- 设计到文件路径的位置，空格和中文字符可能会导致执行错误，请尽量避免使用。
- XML里面的文件路径注意使用\或者使用/。
- XML的语法可以自行搜寻，修改后的温度不能存在语法错误。
- 已知XML注释会导致菜单加载错误，请不用在XML文档中使用注释。

Yongsheng.guo@ansys.com

- XML文件编辑保持后，Toolbox会自动重新加载。

1.6 Exe文件的添加

```
<SubMenu Type="MenuItem" Name="Notepad" ExecuteType="EXE" Path="Notepad.exe" Arguments =""/>
```

属性定义如下：

- Name：菜单显示的名称
- ExecuteType： "EXE"
- Path：可执行文件路径。如果这里指定文档，且系统有默认执行程序，也可以顺利打开，比如Path指定xxx.docx文档。
- Arguments: 传递参数，多个参数以空格隔开。

1.7 Python脚本的添加

```
<SubMenu Type="MenuItem" Name="QuitAedt" ExecuteType="Python" Path="$UserLib/Desktop/Close.py" EntryFunc="ForceQuitAedt" Arguments ="" PythonPath=""></SubMenu>
```

属性定义如下：

- Name：菜单显示的名称
- ExecuteType：Python
- Path: 脚本的路径。可以使用绝对路径值。如果使用相对路径，可以使用″$+目录″的形式引入当面目录下的文件夹。
- Arguments: 可选，允许传递参数给可执行脚本，多个参数以空格隔开。
- PythonPath： 可选，可以指定Python的执行路径，特别是存在多个版本时可以按照路径区分版本。省略时会从Path变量中查找Python执行文件。
- EntryFunc： 可选，针对Python，IronPython指定运行脚本的入口函数，默认为″main"函数（可以省略），如果为其它函数则需要指定函数名称。

1.8 IronPython脚本的添加

```
<SubMenu Type="MenuItem" Name="QuitAedt" ExecuteType="Python" Path="$UserLib/Desktop/Close.py" EntryFunc="ForceQuitAedt" Arguments ="" PythonPath=""></SubMenu>
```

属性定义如下：

- Name：菜单显示的名称
- ExecuteType：Python
- Path: 脚本的路径。可以使用绝对路径值。如果使用相对路径，可以使用″$+目录″的形式引入当面目录下的文件夹。
- Arguments: 可选，允许传递参数给可执行脚本，多个参数以空格隔开。
- PythonPath： 可选，可以指定Python的执行路径，特别是存在多个版本时可以按照路径区分版本。省略时会从Path变量中查找Python执行文件。
- EntryFunc： 可选，针对Python，IronPython指定运行脚本的入口函数，默认为″main"函数（可以省略），如果为其它函数则需要指定函数名称。

1.9 python脚本的编写

python脚本可以按照正常的模式进行开发，将需要运行到功能包装成一个函数，最终在菜单的xml里面指定未入口函数即可。
默认的入口函数未mian()函数，建议使用main()函数作为入口。

下面的案例中，python文件定义了多个函数，可以在xml里面分别指定不同的函数作为入口函数，完成不同的功能。
另外也可以通过传递不同的Arguments来完成类似的功能。

```
import sys,os

Module = sys.modules['__main__']
if hasattr(Module, "oDesktop"):
    oDesktop = getattr(Module, "oDesktop")
else:
```


Yongsheng.guo@ansys.com

```
    raise Exception("oDesktop intial error.")


def closeAndSave():
    oProject = oDesktop.GetActiveProject()
    oProject.Save()
    oProject.Close()

def closeNotSave():
    oProject = oDesktop.GetActiveProject()
    # oProject.Save()
    oProject.Close() #Unsaved changes will be lost.

def closeAllProjectWithSave():
    projects = oDesktop.GetProjects()
    for oProject in projects:
        oProject.Save()
        oProject.Close()

def closeAllProjectWithoutSave():
    projects = oDesktop.GetProjects()
    for oProject in projects:
        # oProject.Save()
        oProject.Close() #Unsaved changes will be lost.

def ForceQuitAedt():
    oDesktop.QuitApplication()

def Reload():
    oProject = oDesktop.GetActiveProject()
    aedtPath = os.path.join(oProject.GetPath(),oProject.GetName()+".aedt")
    oProject.Close()
    print("Reload aedt:%s"%aedtPath)
    oDesktop.OpenProject(aedtPath)
```

XML菜单配置

```
    <SubMenu Name="Close">
        <SubMenu Type="MenuItem" Name="QuitAedt" ExecuteType="Python" Path="$UserLib/Desktop/Close.py" EntryFunc="ForceQuitAedt"></SubMenu>
        <SubMenu Type="MenuItem" Name="CloseProjectAndSave" ExecuteType="Python" Path="$UserLib/Desktop/Close.py" EntryFunc="closeAndSave"></SubMenu>
        <SubMenu Type="MenuItem" Name="CloseProjectNotSave" ExecuteType="Python" Path="$UserLib/Desktop/Close.py" EntryFunc="closeNotSave"></SubMenu>
        <SubMenu Type="MenuItem" Name="CloseAllProjectWithSave" ExecuteType="Python" Path="$UserLib/Desktop/Close.py" EntryFunc="closeAllProjectWithSave"></
SubMenu>
        <SubMenu Type="MenuItem" Name="CloseAllProjectWithoutSave" ExecuteType="Python" Path="$UserLib/Desktop/Close.py"
EntryFunc="closeAllProjectWithoutSave"></SubMenu>
        <SubMenu Type="MenuItem" Name="ReloadProject" ExecuteType="IronPython" Path="$UserLib/Desktop/Close.py" EntryFunc="Reload"></SubMenu>
    </SubMenu>
```


Yongsheng.guo@ansys.com

2. ExportToHfssWithNets

2.1 菜单位置

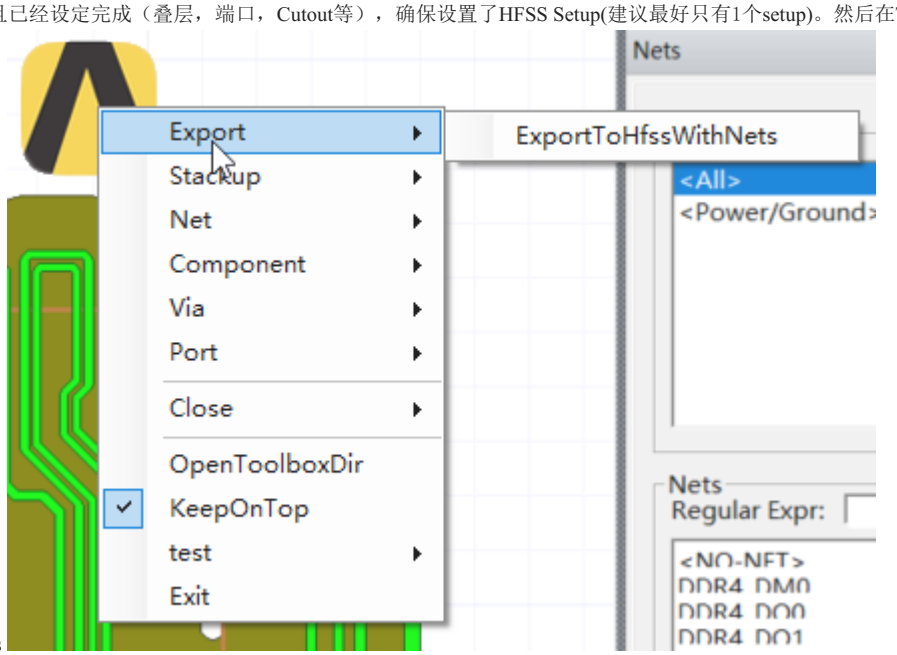
Export->ExportToHfssWithNets

2.2 功能描述

将3D Layout中的3D模型导出到HFSS后，可以按照Net对物体进行分组，这样更便于查看和设定物体。

2.3 操作说明

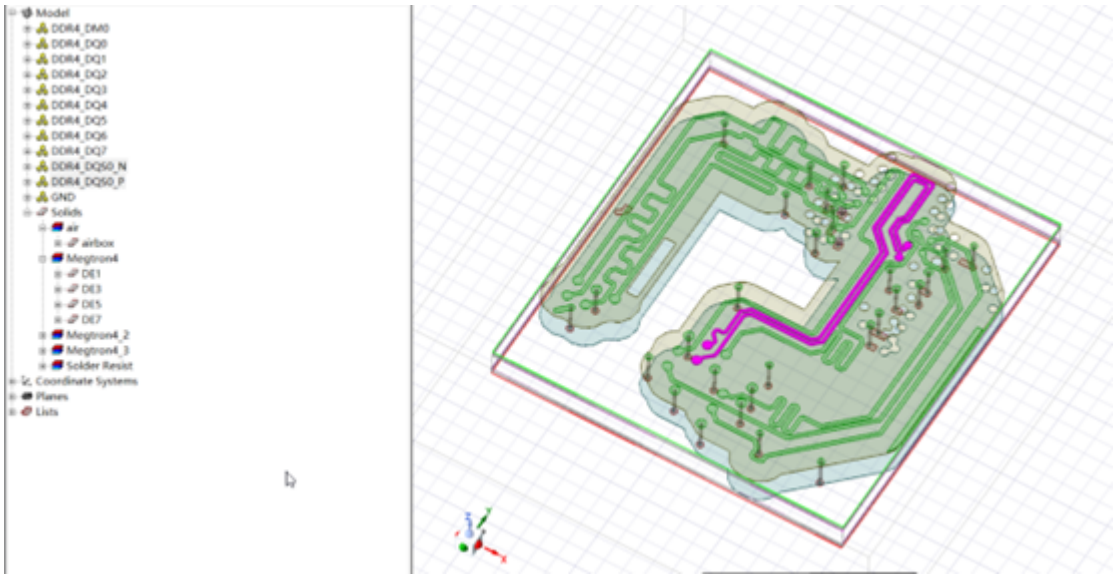
确保3D Layout工程处于选中状态，并且已经设定完成（叠层，端口，Cutout等），确保设置了HFSS Setup(建议最好只有1个setup)。然后在Toolbox上右键，选择Export->ExportToHfssWithNets



2.4 完成效果

等待脚本执行完成，最终效果如下：

Yongsheng.guo@ansys.com



联系作者 yongsheng.guo@ansys.com

3. ImportCSVStackup

3.1 菜单位置

Stackup->ImportCSVStackup

3.2 功能描述

通过CSV文件快速导入加工厂给定的叠层。可以把工厂提供的xlsx文件按照提供的CSV模板格式输出成文件做输入即可。列的顺序不限，不在定义内的列会被忽略。

3.3 操作说明

需要按照提供的CSV模板填写叠层信息。然后在Toolbox上右键，选择Stackup->ImportCSVStackup，选择已经填写好的CSV叠层模板，进行导入。
在ImportCSVStackup.py目录下提供了"CsvStackupTemp_full.csv" 和 "CsvStackupTemp_simple.csv"两个模板可供使用。

3.4 模板格式

Layer	Type	Thickness(mil)	Cond	DK	DF	Roughn
UNNAMED_000	dielectric	0		3.7	0.15	0.5um
TOP	signal	1.9		3.7	0.15	0.5um
UNNAMED_002	dielectric	2.65		3.5	0.004	0.5um
PWR	signal	1.3		3.5	0.004	0.5um
UNNAMED_004	dielectric	50		3.5	0.004	0.5um
LYR_1	signal	1.2		3.5	0.004	0.5um
UNNAMED_006	dielectric	8		3.5	0.004	0.5um
LYR_2	signal	1.2		3.5	0.004	0.5um
UNNAMED_008	dielectric	8		3.5	0.004	0.5um
GND	signal	1.3		3.5	0.004	0.5um
UNNAMED_010	dielectric	2.65		3.5	0.004	0.5um
BOTTOM	signal	1.9		3.7	0.15	0.5um
UNNAMED_012	dielectric	0		3.7	0.15	0.5um

说明： 1. Layer, Type,Thickness必须指定，未指定的行将被忽略。 2. Cond, DK, DF如果缺失或者为给定数值，默认会使用Copper和FR4_epoxy替代 3. Thickness可以使用()指定单位，或者在数值中直接使用单位。 4. Roughness为可选属性，Groiss格式：0.5um, huray格式：0.5um;2.9

3.5 可选属性

- Thickness：厚度，需要带单位。或者在标题上使用全局单位，比如Thickness(mil) or Thickness(mm)
- Material：指定系统库里面的材料名称，比如Copper，此时Cond/DK/DF被忽略
- FillMaterial:指定系统库里面的材料名称，比如FR-4，此时Cond/DK/DF被忽略
- Cond/DK/DF 如果指定，且未指定Material或FillMaterial，自动根据参数生成新材料

Yongsheng.guo@ansys.com

- Roughness：指定粗糙度，Groiss格式0. 5um, huray格式：0. 5um;2.9 or 0.5um,2.9
- EtchFactor: 蚀刻系数，正值为Top蚀刻，负值为Bottom蚀刻

3.6 注意事项

- CSV中的金属层数量必须和PCB里面的相同，否则无法更新。
- CSV的介质层数量不要求和PCB里面的相同，如果介质层数不同，CSV的介质层强制覆盖PCB中的介质层。

联系作者 yongsheng.guo@ansys.com

4. AutoXNet

4.1 菜单位置

Net->AutoXNet

4.2 功能描述

当信号线经过电容并且一侧的网络没有使用命名（网络为数字或者随机名称），本脚本可以根据一侧已经命名的网络，对另外一侧进行命名，新命名的网络会根据RLC的不同添加_C,_R,_L的标志。（当前默认值对Cap）

4.3 操作说明

确保3D Layout工程处于选中状态，然后在Toolbox上右键，选择 Net->AutoXNet. 脚本会自动在当前激活的AEDT里面执行，检测电容两侧的网络，对于未命名的网络进行命名。

4.4 注意事项

- 脚本会直接跳过3D Layout里面定义的Power/GND网络。
- Cap需要分配到Component的Capacitors里面

联系作者 yongsheng.guo@ansys.com

5. deleteInvalidRLC

5.1 菜单位置

Component->deleteInvalidRLC

5.2 功能描述

删除只存在1个有效网络的RLC器件。对Layout进行Cutout和网络删除后，部分RLC的网络被删除，处于无效状态，次命令可以清除无效的RLC器件。

5.3 操作说明

确保3D Layout工程处于选中状态，然后在Toolbox上右键，选择 Component->deleteInvalidRLC. 脚本会自动在当前激活的AEDT里面执行，检测RLC的Net状态，并删除无效的网络。

联系作者 yongsheng.guo@ansys.com

6. AutoBackdrill

6.1 菜单位置

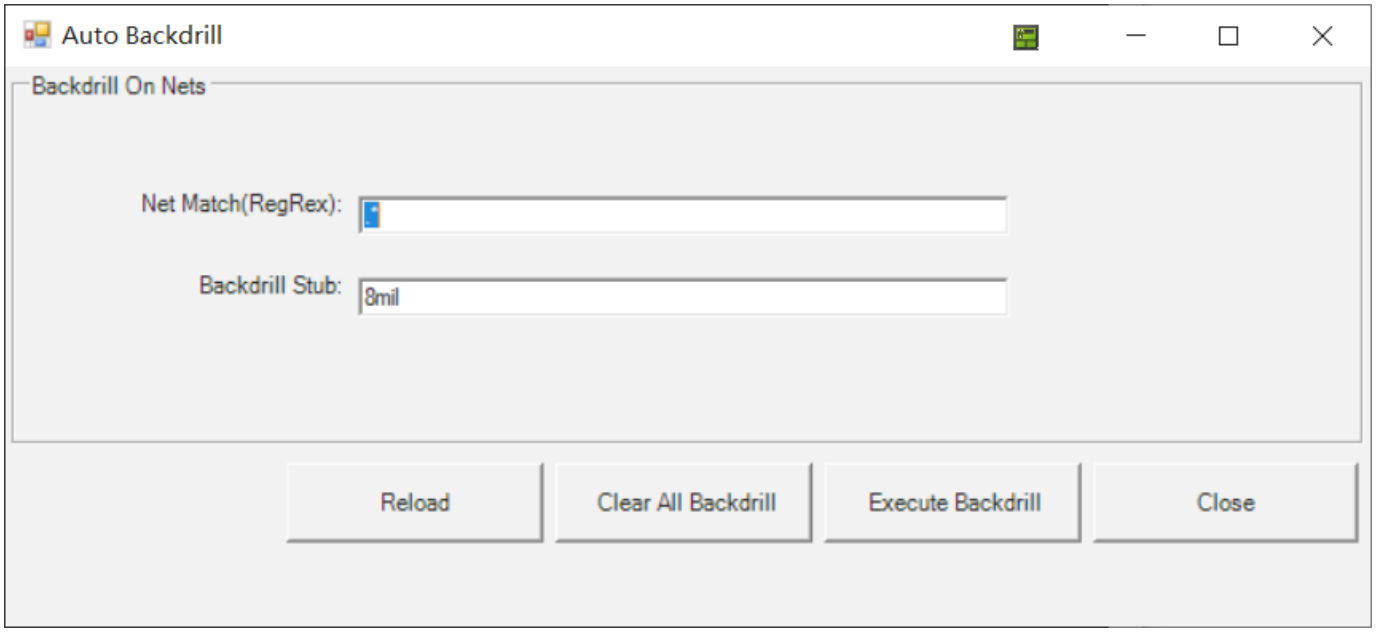
Via->AutoBackdrill

6.2 功能描述

自动设定过孔的背钻，支持指定网络名称和Stub长度。仅支持2023R1及之后的版本。

6.3 操作说明

确保3D Layout工程处于选中状态，然后在Toolbox上右键，选择 Via->AutoBackdrill. 会弹出AutoBackdrill设定对话框



- Net Match(Regrex)：使用正则表达式匹配需要背钻的网络，默认.*匹配所有网络。
- Backdrill Stub: 背钻精度，默认8mil

6.4 Net Match(Regrex)

- 正则表达式使用. 匹配任意多字符，比如使用*DQ*.匹配以DQ开头的Nets.
- 正则表达式的写法请参考 https://www.runoob.com/regexp/regexp-syntax.html

6.5 注意事项

- 支持总线的书写方法，比如使用DQ[15:0]匹配 DQ0-DQ15的网络。（注：这种写法并不是Regrex的一部分）

联系作者 yongsheng.guo@ansys.com