

Abaqus-Python API Operation Manual

2022年5月13日 21:07

Project deployment, Please refer to <https://github.com/JiaoranWang/Python-Abaqus.git>

The original .PDF of this file is here:

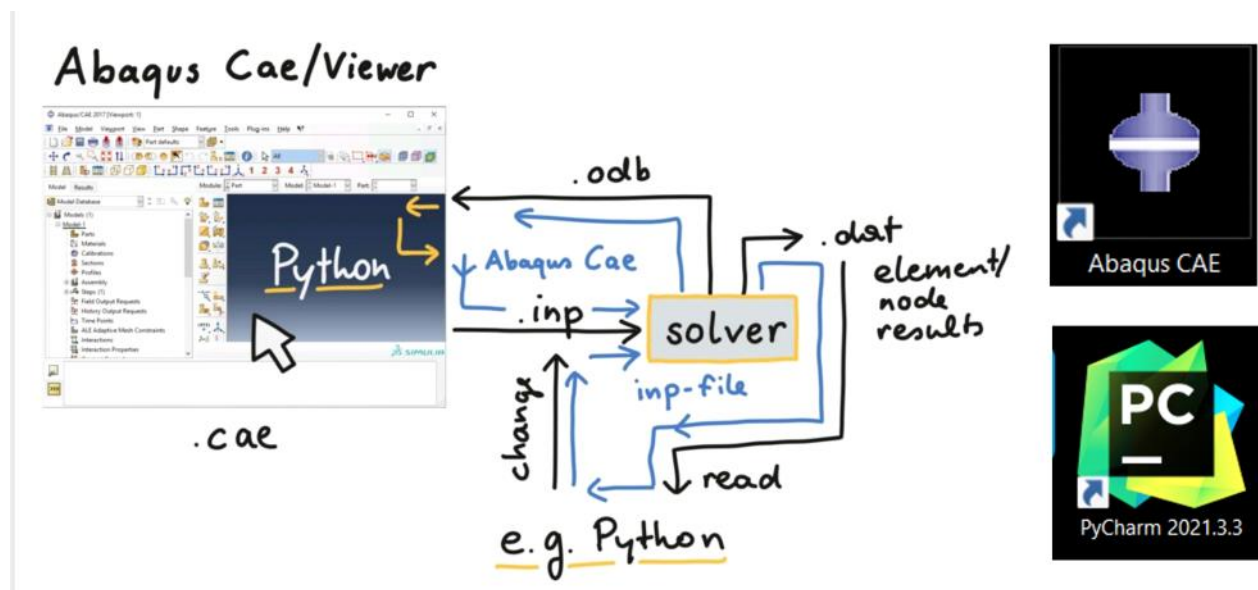
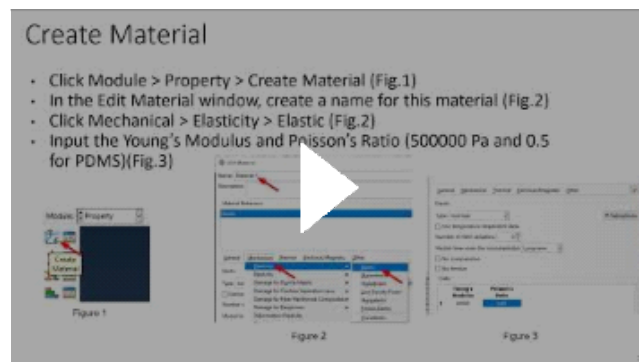
The abaqus modeling file mentioned is here:

(.ppt):

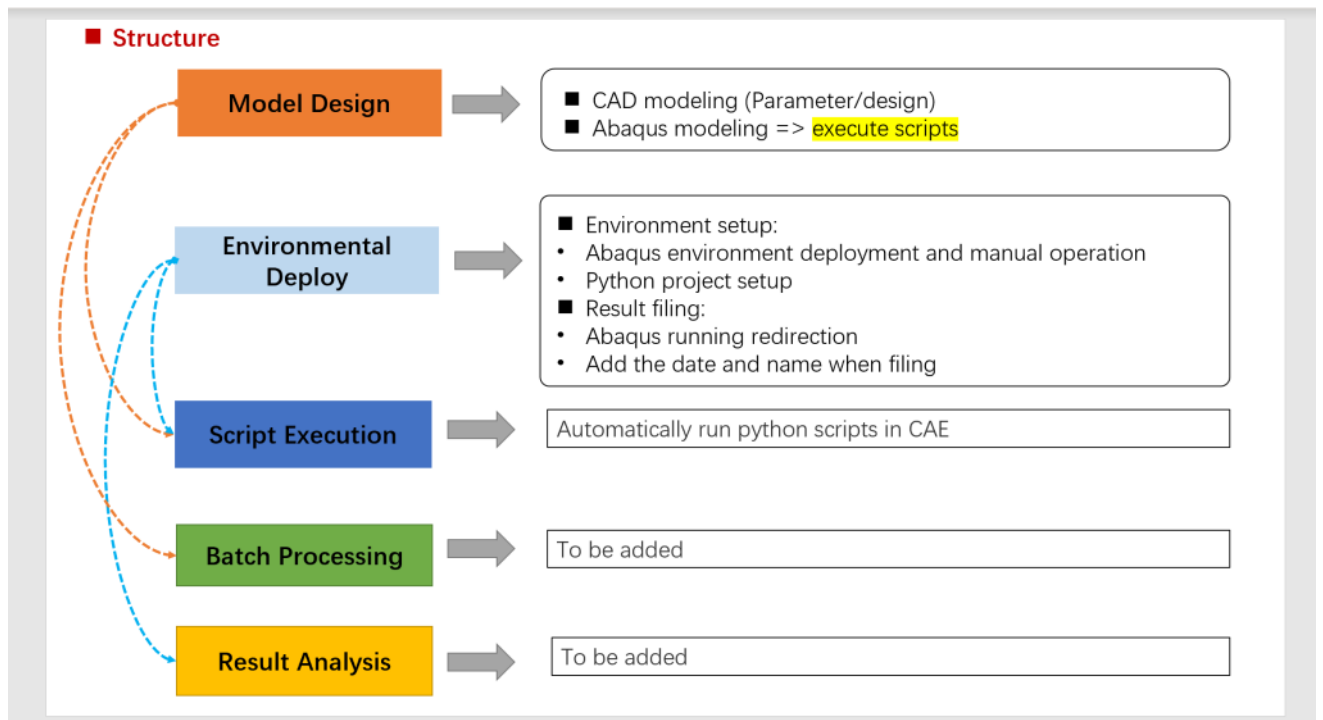
<https://github.com/JiaoranWang/Python-Abaqus/blob/master/Python-Abaqus%20API%20Project/Pics/3%20abaqus%20tutorial.pdf>

(.mp4)

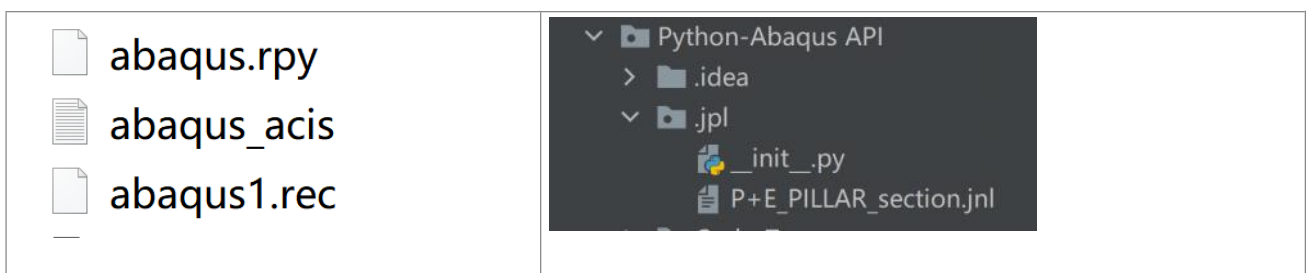
[Abaqus Simulation Tutorial](#)



1.intro



Abaqus can automatically generate Python script files with the extension.rpy, it's automatically tracking all the operations in the CAE interface. When the task ends, the scripts are saved into .jnl, you can run the file to rebuild the model



The "abaqus.rpy" can be automatically updated according to action (checked in notebook++ and Pycharm) By changing the parameters and process in it, we can obtain a new model with new job and analysis.

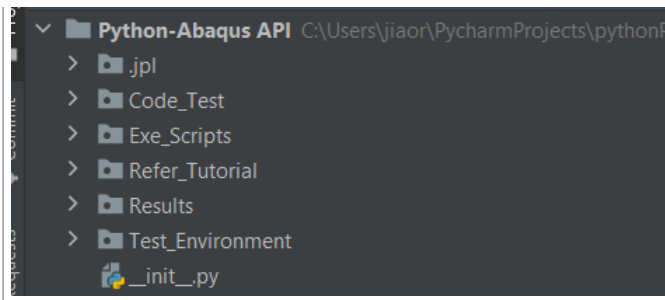
To be noticed that CAE can only run the plain python scripts in its run shell, which means that:

- 1) Python scripts that have judgment statements (because prespace changes are not recognized)
- and 2) reference functions defined in the same file (CAE does not redirect to relative pathlookup) are not allowed to be imported into CAE to run
- => 3) Any references, function definitions and variable inputs need to be in the same script

2. Running Python scripts in Abaqus

2.1 Function structure and specification

For now, we have the following scripts:



Please see the following pic

■ Function structure and specification

C) Execute scripts (change it from Abaqus.rpy)

D) Where you put results

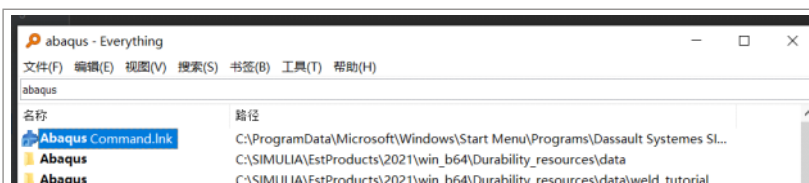
B) This file change the Abaqus running directory, so the running and results can be redirect to a desire path. E.g., in the "Results" folder
**** Need to be added into the front of Exe_1.py (the script you want to run)

A) To automatically or manually run the scripts using [Abaqus Command] Python feeds the .bat file into the

1.Environment

1.1 Abaqus Environment: How do I automatically run scripts in Abaqus

Firstly we need to check where the 'Abaqus Command' window are.



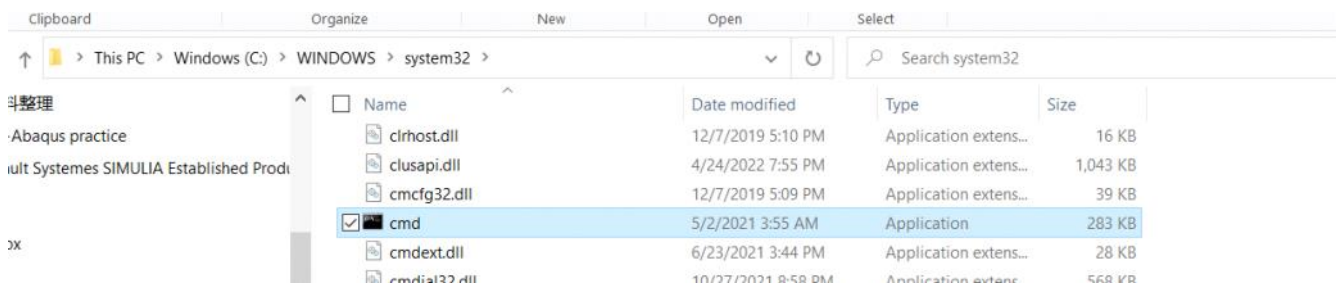
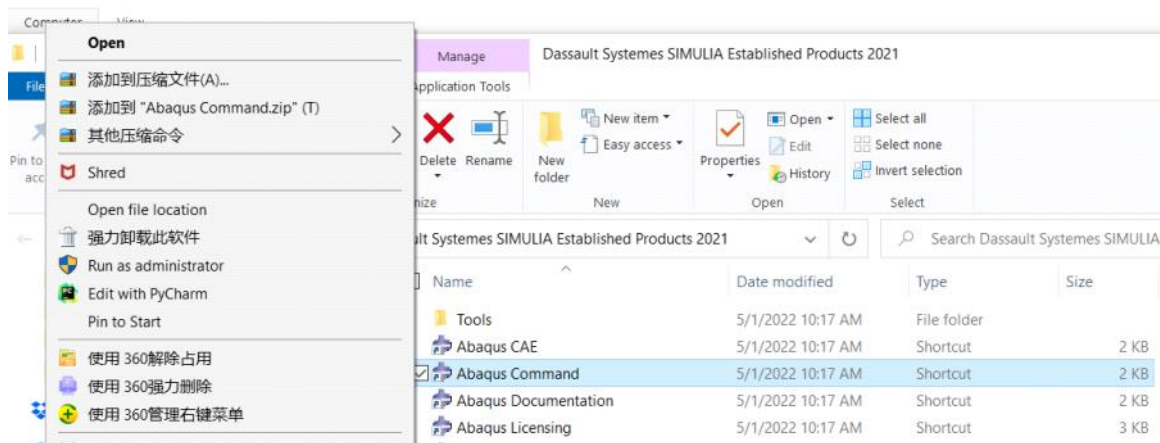
C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Dassault Systemes SIMULIA Established Products 2021

My path is in C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Dassault Systemes SIMULIA Established Products 2021 => **"Abaqus Command.Ink"**

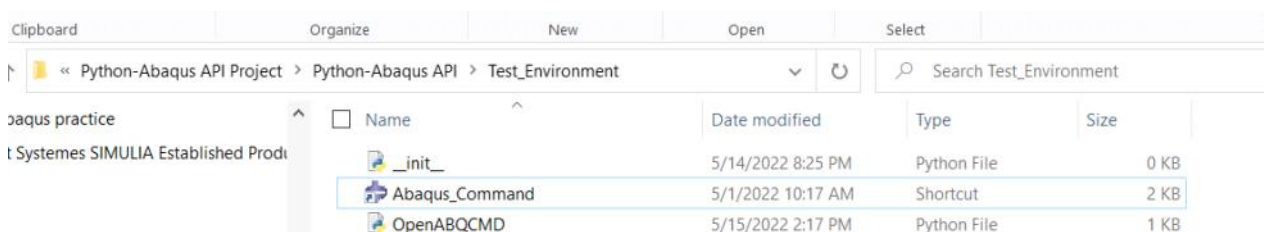
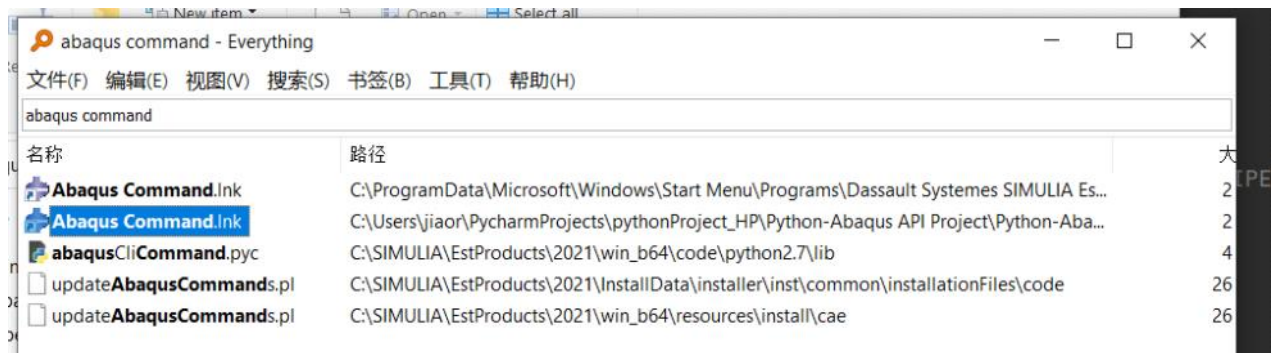
The .lnk is the suffix for shortcut, to trace back to the original file, we can right click, then click on the 'Open file location'

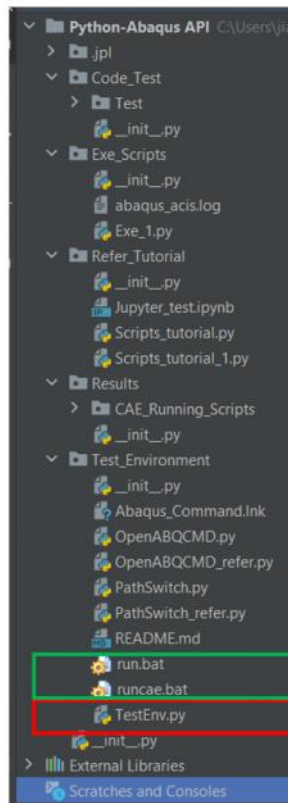
(How do I find the normal location of the shortcut?)

https://m33.wiki/help/windows_shortcut_original_file_location.html)



The original file is a CMD, so we can open it in Python and type input by feeding .bat file
 [Strongly suggestion] Since the .lnk could function well, we can put it directly into the folder where we calling it to avoid path changing (see the following pic)





■ Open [Abaqus Command.lnk] and run the .py script

```
'''### Open Abaqus Command window then Run Python Script ###'''
import subprocess
import sys
import time

get_input = input("Please input: Start the CAE with scripts? Y/N")

if get_input.strip().upper() == "Y":
    # do not need to be trouble go directly to run.bat
    p = subprocess.Popen(["Abaqus_Command.lnk", "start runcae.bat"], shell=True, stdout=subprocess.PIPE,
                        stderr=subprocess.STDOUT)
    print('--- CMD->.bat opened! sleep 10 second... ---')
    output = p.stdout.readline()
    print('output = ', output)

    # p.wait() # wait for the process to finish
    # p.communicate("cd C:\sa") # input
    time.sleep(10) # wait for 10 seconds
    print('--- CMD process END ---')
    sys.exit(0) # exit
else:
    p = subprocess.Popen(["cmd", "/k", "start run.bat"], shell=True, stdout=subprocess.PIPE) # do not need to be
    # do not need to be trouble go directly to run.bat
```

```
cd ..
cd C:\Users\jiaor\PycharmProjects\pythonProject_HP\Python-Abaqus API Project\Python-Abaqus API\Exe_Scripts
dir
abaqus cae script=Exe_1.py
//
abaqus cae script=test.py
```

```
'''### Open Abaqus Command window then Run Python Script ###'''
import subprocess
import sys
import time

get_input = input("Please input: Start the CAE with scripts? Y/N")

if get_input.strip().upper() == "Y":
    # do not need to be trouble go directly to run.bat
    p = subprocess.Popen(["Abaqus_Command.lnk", "start runcae.bat"], shell=True, stdout=subprocess.PIPE,
                        stderr=subprocess.STDOUT)
    print('--- CMD->.bat opened! sleep 10 second... ---')
    output = p.stdout.readline()
    print('output = ', output)

    # p.wait() # wait for the process to finish
    # p.communicate("cd C:\sa") # input
    time.sleep(10) # wait for 10 seconds
    print('--- CMD process END ---')
    sys.exit(0) # exit
else:
    p = subprocess.Popen(["cmd", "/k", "start run.bat"], shell=True, stdout=subprocess.PIPE) # do not need to be
    # do not need to be trouble go directly to run.bat
```

Enter the Command in the Abaqus Command window:

"

cd C:\Users\jiaor\PycharmProjects\pythonProject_HP\Python-Abaqus API Project\Python-Abaqus API\Exe_Scripts

abaqus cae script=Exe_1.py

"

I put them into a .bat file


```
p = subprocess.Popen(["Abaqus_Command.lnk", "start runcae.bat"], shell=True, stdout=subprocess.PIPE)

cd ..
cd C:\Users\jiaor\PycharmProjects\pythonProject_HP\Python-Abaqus API Project\Python-Abaqus API\Exe_Scripts
cd Exe_Scripts
dir
abaqus cae script=Exe_1.py
//
abaqus cae script=test.py
```

To directly put the path into abaqus cae script could not work:

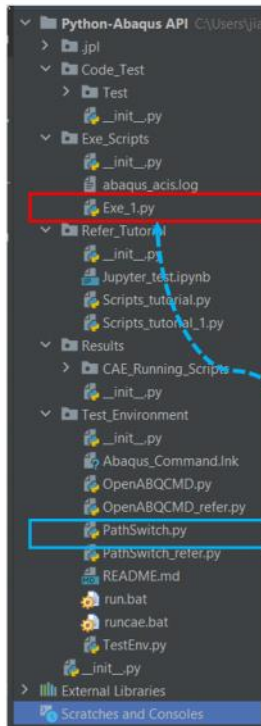
~~abaqus cae script=C:\Users\jiaor\PycharmProjects\pythonProject_HP\Python-Abaqus API Project\Python-Abaqus API\Exe_Scripts\Exe_1.py~~

Execute Scripts

■ PathSwitch

```
### [Abaqus] Change/set the work dictionary and results ###
import os
import sys
from datetime import datetime
# Acquire the time
time = datetime.now().strftime("%Y%m%d%H%M")
# Acquire Script Name
name = os.path.basename(sys.argv[0])
name = name[:-3]
# change the work to folder
folder = os.path.abspath(os.path.join(os.getcwd(), "..")+"\\Results\\"+'Test_'+time+'_'+name)
# make and change the work to folder
os.makedirs(folder)
os.chdir(folder)
# -----#
```

Folder name format: Test_mmddtt_xxxxxx(.py script name)



■ Execute Script

```
time = datetime.now().strftime("%Y%m%d%H%M%S")
# Acquire Script Name
name = os.path.basename(sys.argv[0])
name = name[:-3]
# change the work to folder
folder = os.path.abspath(os.path.join(os.getcwd(), "...")) + "\\Results\\" + "Test_" + time + "_" + name
# make and change the work to folder
os.makedirs(folder)
os.chdir(folder)
# -----
```

```
def Create_Part_3D_Cylinder(radius, length, thickness, part, model):
    s1 = mdb.models[model].ConstrainedSketch(name='__profile__', sheetSize=200.0)
    p, v, d, c = s1.geometry, s1.vertices, s1.dimensions, s1.constraints
    s1.setPrimaryObject(notice=STANDALONE)
    s1.CircleByCenterPerimeter(center=(0.0, 0.0), point1=(radius, 0.0))
    s1.CircleByCenterPerimeter(center=(0.0, 0.0), point1=(radius - thickness, 0.0))
    p = mdb.models[model].Part(name=part, dimensionality=THREE_D, type=DEFORMABLE_BODY)
    p = mdb.models[model].parts[part]
    p.BaseSolidExtrude(sketch=s1, depth=length)
    s1.unsetPrimaryObject()
    p = mdb.models[model].parts[part]
    del mdb.models[model].sketches['__profile__']

def Create_Datum_Plane_by_Principal(type_plane, part, model, offset_plane):
    p = mdb.models[model].parts[part]
```

```
Create_Set_Edge(myRadius = mythickness / 2.0, 0.0, myLength / 2.0, myString, myPart, "Set-3DGA-Point")
# Create_Boundary_Condition_by_Instance(myString, "Cylinder-1", "Set-3DGA-Point", "BC-Imperfection", "Step-1", 0.0, UNSET, 0)
Create_Set_Vertice(0.0, myRadius, myLength / 2.0, myString, myPart, "SPLA_Point")
# Create_SPLAOnString, "Cylinder-1", "SPLA_Point", "Perturbation_Load", "Step-1", -1000.0)
# Create_Pressure_Load(myString, "Cylinder-1", "Internal_Pressure", "Step-1", "Internal_Surface", 10)
```

[Abaqus]Change/settheworkdictionaryandresults