Python GUI using Gooey

# Some background

* Gooey library can be used to create an modern UI on top of a Python script/program
* This works for Python programs which accepts certain mandatory or optional inputs from user and runs to completion doing what it is supposed to do.
* It will not work if one program has to handover the reigns to another program and so on.
* Here is the github page of Gooey - <https://github.com/chriskiehl/Gooey>
* It can be installed using pip install gooey
* Gooey provides a decorator which automatically adds UI to a python program.

# What is needed ?

* The script should use argparser to parse the inputs.
* GooeyParser can be used instead of argparser if required.
* The github page has details on the various options that can be passed to the decorator function.
* An example is provided in the next section.

# An example

* Below is a simple program which accepts certain inputs and prints it out via another function.
* It accepts the following inputs:
  + Machine Name which is a text input.
  + Module which is a choice list with a default value
  + User
  + Password (password should not show up in UI)
* run\_engine() is the main function
* print\_func() is another function called by main function.
* @Gooey (…) is the decorator function called on the main function.
* This decorator accepts certain parameters which will change the behavior of the UI.
  + For example: program\_name, progress\_regex, hide\_message are used here.
  + program name is for showing the name on the UI.
  + progress\_regex and hide\_message are used to show a progress bar in the UI based on certain print outputs in the script.
  + There are various other options that can be used.
* The main function has used GooeyParser to add arguments. All arguments are mandatory in this example.

from gooey import Gooey, GooeyParser

def print\_func(inputvar, inputtext):

print(inputvar + ' value is : '+ inputtext+ '\n')

@Gooey(program\_name='Test Utility',

progress\_regex=r"^progress: (\d+)%$", hide\_progress\_msg=True)

def run\_engine():

module\_list=['PIM','MFG','ADHOC']

default\_module=module\_list[0]

my\_parser = GooeyParser(description='Run Diagnostic queries on a Machine')

my\_parser.add\_argument('MACHINENAME', metavar='Enter the Machine Name', action='store', type=str)

my\_parser.add\_argument('MODULE', metavar='Choose the Module', action='store', type=str, choices=module\_list, default=default\_module )

my\_parser.add\_argument('USER', metavar='Enter the username ', action='store', type=str )

my\_parser.add\_argument('PWD', metavar='Enter the password ', action='store', widget='PasswordField' )

args = my\_parser.parse\_args()

print('progress: 20%')

print\_func('Machine Name',args.MACHINENAME)

print('progress: 40%')

print\_func('Module Name',args.MODULE)

print\_func('User Name',args.USER)

print('progress: 80%')

print\_func('Password', '\*\*\*\*\*\*\*\* Password will not be shown ')

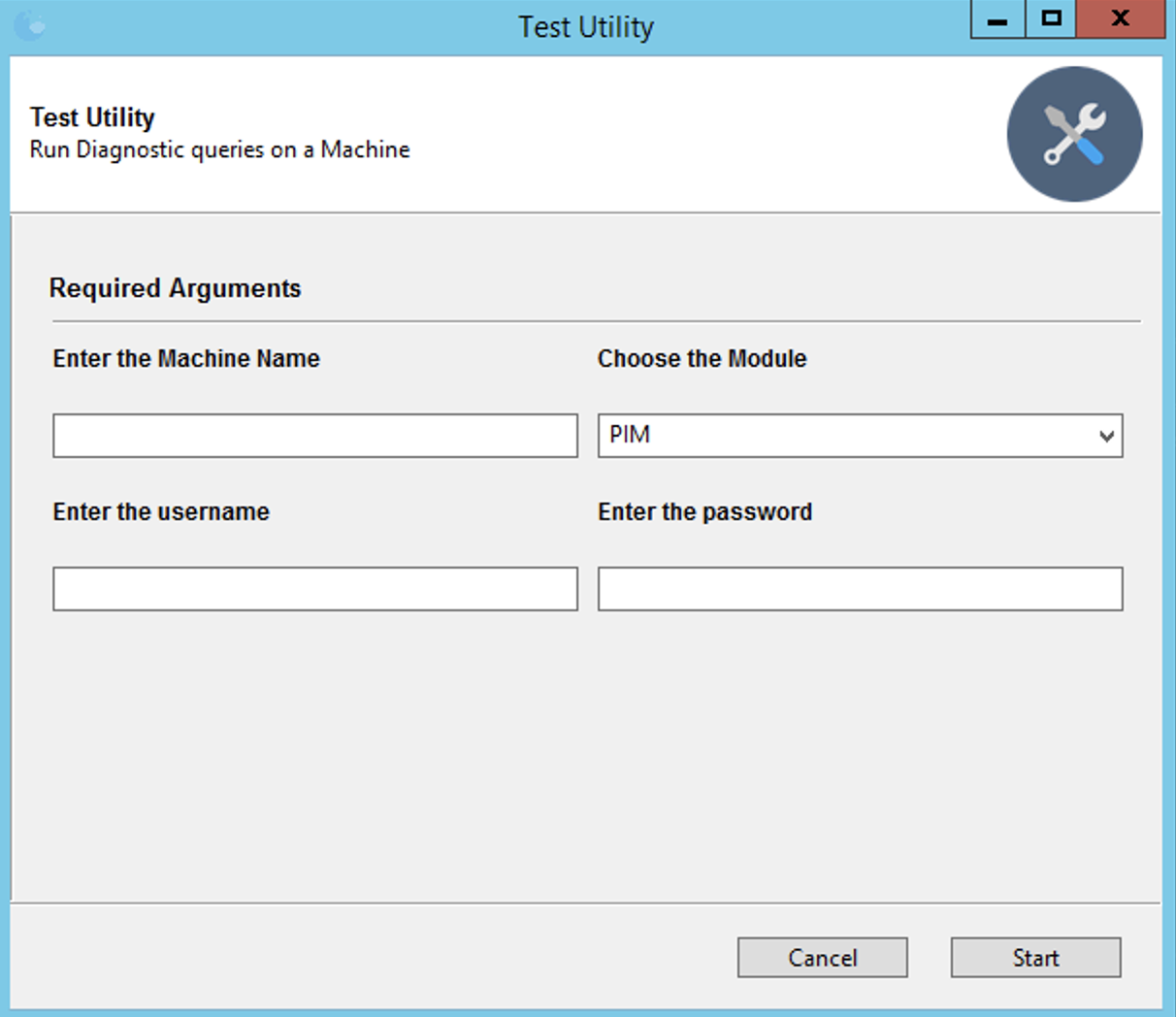
print('progress: 100%')

if \_\_name\_\_ == '\_\_main\_\_':

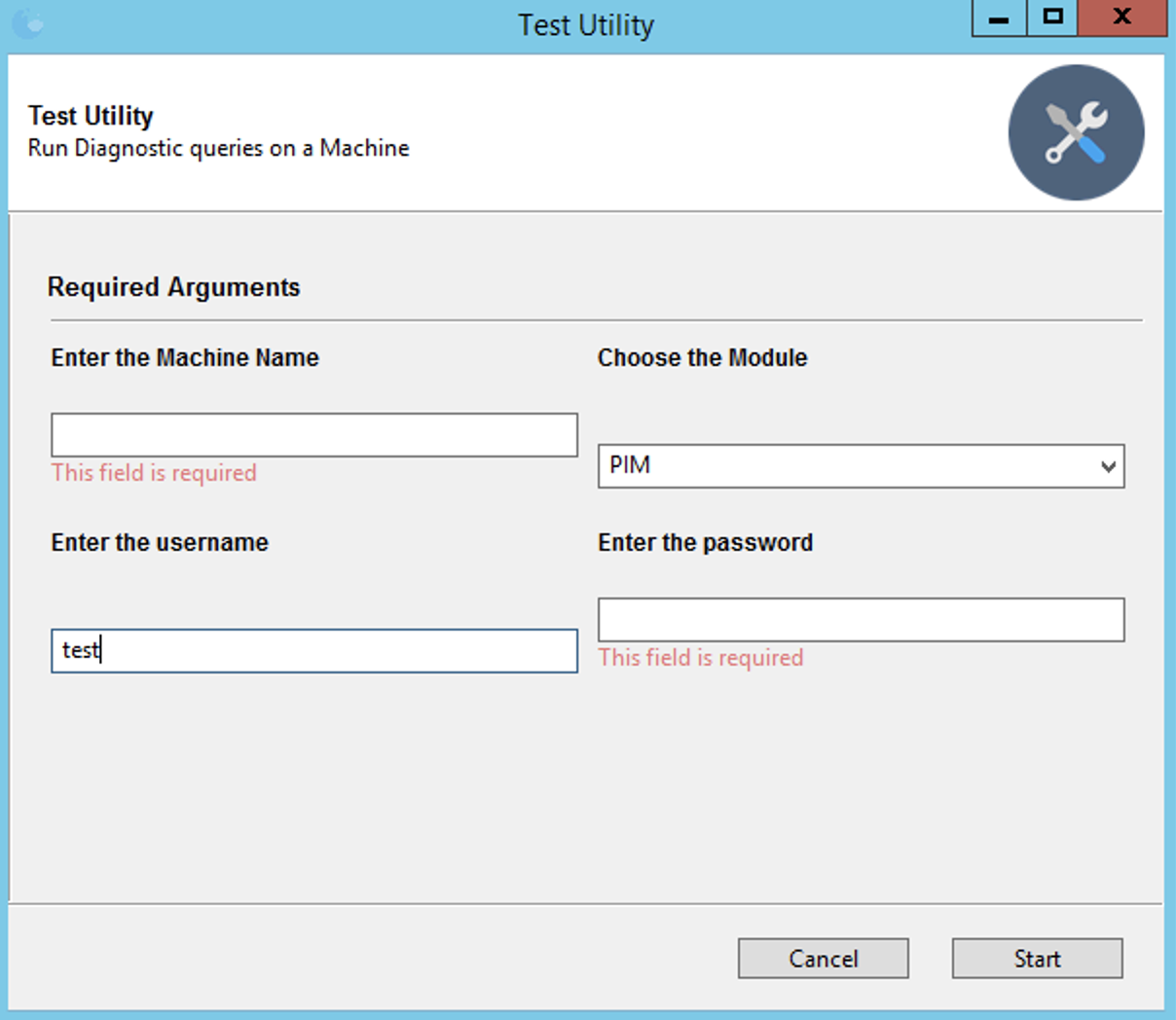
run\_engine()

# Output

* Startup screen -



* Enter the details and Press Enter.
* If values are not entered for mandatory inputs, the UI throws error as shown below.



* Once the values are entered and Start button is pressed, the program is invoked.
* Once the program completes, the UI shows the outputs in “status” window as shown.

