RunPy - Frequently Asked Qustion (for end users)

1. What is RunPy ?

RunPy is an application launcher for Python Script. It runs on 64bits windows (requires 8G RAM) and have the following features:

a. it can be distributed with portable, embeded python system file i.e. the end users need not install python in their computer;
b. It enable user to enter input data via windows graphical interface.

With proper modification of your python script and configuration (RunPy.cfg), RunPy can launch and run any python script.

2. What is RunPy.cfg (configuration file)?

Runpy.cfg is a text file which contains the specifiction for each python application that to be launched by Runpy program.

The format of each item in the configuration file is as follows

```
"Name":{["DESC":"description","IP":[{"var1", value1," var2", value2,  …],"OP":["file1", "file2", …]}
```

"Name" - Name of the python file to run (exclude the file extension '.py');

"DESC" - Description to be shown in the application list;

“IP”: Input variables required by the python program. var1 is the description to be displayed on screen, value1 is the default value for variable 1, and so on. The current version of RunPy.exe supports up to three variables input;

“OP”: Name of files (e.g. images, pdf) that will be created by the python program and displayed or saved in RunPy application.

Runpy read the configuration file (and update the application list) when it starts. If the Runpy.cfg in _db is deleted, the program will read the configuration data from Runpy.db in the _db folder.

Please note that the section name (DESC, IP, OP) must be in uppercase. Other items (e.g. Name of py file) is not case sensitive. All text must be double quoted. However, default value need not be quoted if it is a numeric data type (to be used for computation)

3. How to add/delete items to the application list

<u>To add an items:</u>
Say we want to add a python program named "sort.py" (this script takes a comma separated value from the user and then convert it into an array).
First, we can use notepad to add the following line to RunPy.cfg:

```
"Sort":{"DESC":"9. Convert a string into array","IP":["CSV string","zz,99,aa"]}
```

(make sure to add a comma between the above new item and the existing items in the file)
Secondly, create python script sort.py, for example :

```
x=var1.value.decode("utf-8")
ls=x.split(',')
print('original: ',x)
print('Unsorted array: ',ls)
print('sorted array: ',sorted(ls))
```

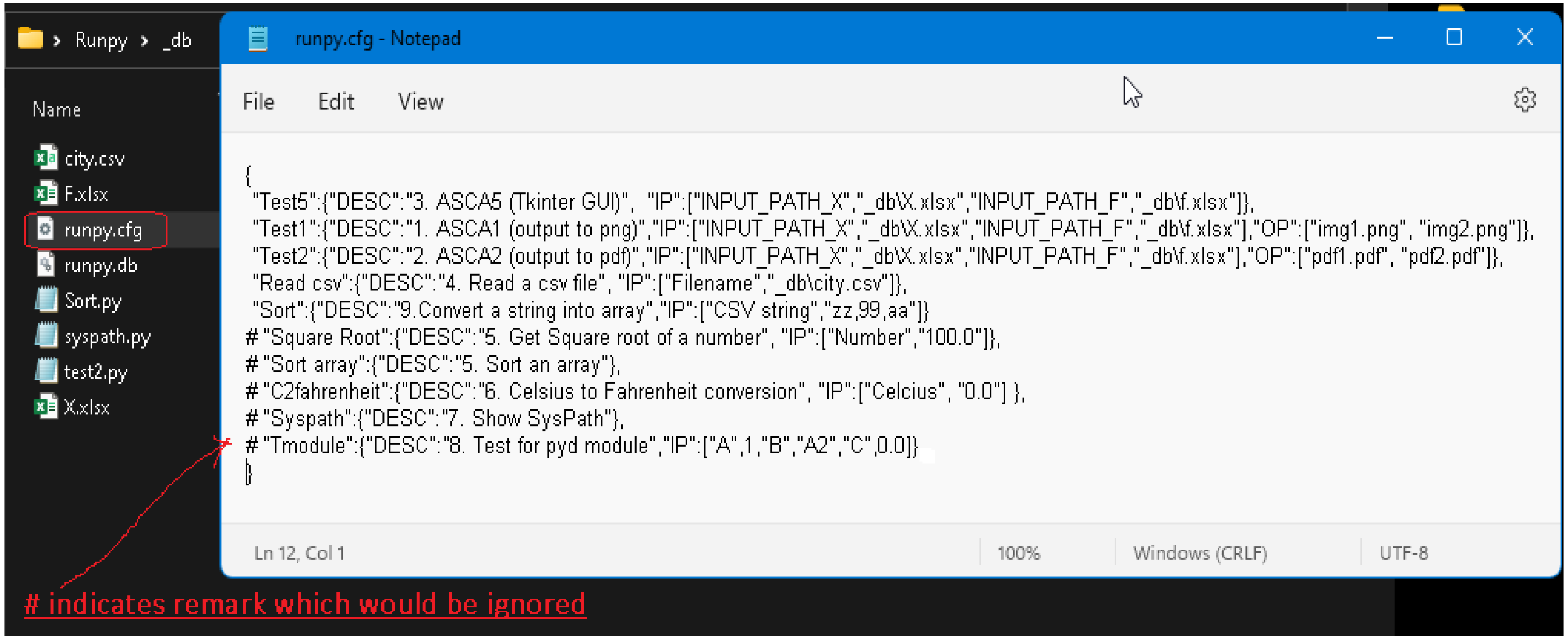
Save the file in _db folder.
Now if we run RunPy.exe, the new item ("Sort") will be appear in the application list. Select 'Sort' from the application list and click run, Runpy will load the sort.py and execute the script.

<u>To delete an item from the application</u>
Just delete the relevant line from RunPy.cfg file.

Sample

```
{
"Test5":{"DESC":"3. ASCA5 (Tkinter GUI)", "IP":["INPUT_PATH_X","_db\X.xlsx","INPUT_PATH_F","_db\f.xlsx"]},
"Test1":{"DESC":"1. ASCA1 (output to png)","IP":["INPUT_PATH_X","_db\X.xlsx","INPUT_PATH_F","_db\f.xlsx"],"OP":["img1.png",
"img2.png"]},
"Test2":{"DESC":"2. ASCA2 (output to pdf)","IP":["INPUT_PATH_X","_db\X.xlsx","INPUT_PATH_F","_db\f.xlsx"],"OP":["pdf1.pdf",
"pdf2.pdf"]},
"Read csv":{"DESC":"4. Read a csv file", "IP":["Filename","_db\city.csv"]},
"Sort":{"DESC":"9.Convert a string into array","IP":["CSV string","zz,99,aa"]}
# "Square Root":{"DESC":"5. Get Square root of a number", "IP":["Number","100.0"]}
}
```

3. How to add/delete items to the application list? (continued)



```
{
 "Test5":{"DESC":"3. ASCA5 (Tkinter GUI)",  "IP":["INPUT_PATH_X","_db\X.xlsx","INPUT_PATH_F","_db\f.xlsx"]},
 "Test1":{"DESC":"1. ASCA1 (output to png)","IP":["INPUT_PATH_X","_db\X.xlsx","INPUT_PATH_F","_db\f.xlsx"],"OP":["img1.png", "img2.png"]},
 "Test2":{"DESC":"2. ASCA2 (output to pdf)","IP":["INPUT_PATH_X","_db\X.xlsx","INPUT_PATH_F","_db\f.xlsx"],"OP":["pdf1.pdf", "pdf2.pdf"]},
 "Read csv":{"DESC":"4. Read a csv file", "IP":["Filename","_db\city.csv"]},
 "Sort":{"DESC":"9.Convert a string into array","IP":["CSV string","zz,99,aa"]}
# "Square Root":{"DESC":"5. Get Square root of a number", "IP":["Number","100.0"]},
# "Sort array":{"DESC":"5. Sort an array"},
# "C2fahrenheit":{"DESC":"6. Celsius to Fahrenheit conversion", "IP":["Celcius", "0.0"] },
# "Syspath":{"DESC":"7. Show SysPath"},
# "Tmodule":{"DESC":"8. Test for pyd module","IP":["A",1,"B","A2","C",0.0]}
}
```

indicates remark which would be ignored

Explanation:-
"Test5" - The python program file is test5.py (stored in _db folder);
"3. ASCA5 (Tkinter GUI)" - Description of test.py that will be shown in the selection box;
"INPUT_PATH_X" - Description of first input variable (var1) default value is _db\X.xlsx;
"INPUT_PATH_F" - Description of second input variable (var2), default value is _db\f.xlsx;
There are only two input variables for Test5.py and there is no output images/file.

"Test1" - The python program file is test1.py. This is similar to Test5 above except that it saves output to img1.png and img2.png (instead of display the images in tk windows)

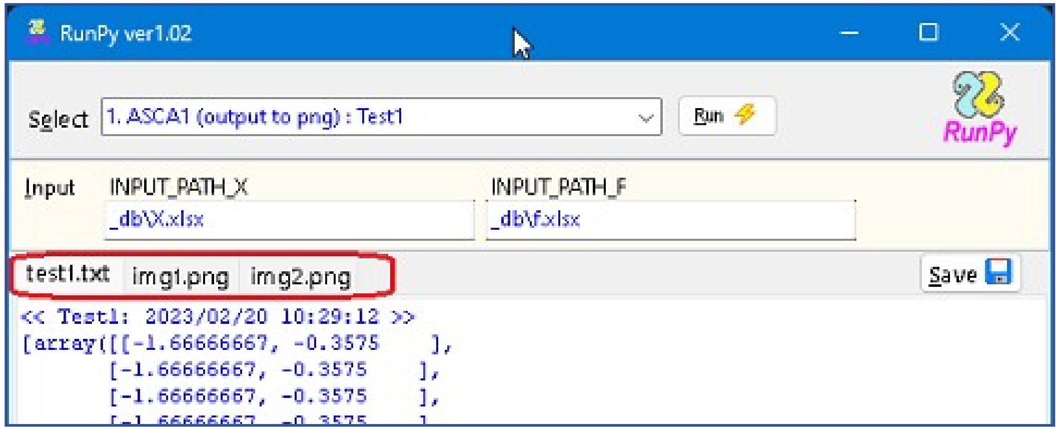
"Test2" - The python program file is test2.py. This is similar to Test2 above except that it saves output to pdf1.pdf and pdf1.pdf.

Line starts with # is indicates that it is comment and will be ignored by RunPy.

There are two output files in the above example. There is actually no limit on number of output files.

4. What does the tab header of indicate?

Each of the tab sheet will represent the output from the python file. The first tab normally is the text output from python program. The rest of the tabs may represent the image, pdf or other type of files output from the selected python application. Note that, for simplicity, RunPy only display the text and png file. Other types of file eg. pdf file would be opened using Windows default application. For example, pdf file may be opened in Adobe reader etc.



Select 1. ASCA1 (output to png) : Test1

Input INPUT_PATH_X _db\X.xlsx INPUT_PATH_F _db\f.xlsx

```
<< Test1: 2023/02/20 10:29:12 >>
[array([[-1.66666667, -0.3575    ],
       [-1.66666667, -0.3575    ],
       [-1.66666667, -0.3575    ],
       [-1.66666667, -0.3575    ]
```

5. How does Runpy affect the coding of a python program?

i. Data input RunPy are accessible in python program via shared objects. As such the pythone file may need to be changed in order to access the share data. Current version of Runpy allows maximum three shared objects, namely var1, var2, and var3. For example, statement in the python script like x = input('Enter your name:') may need to be changed to x=var1.value.decode("utf-8"). Note that string data usually needs to be convert to utf8 format using the decode("utf-8") method.

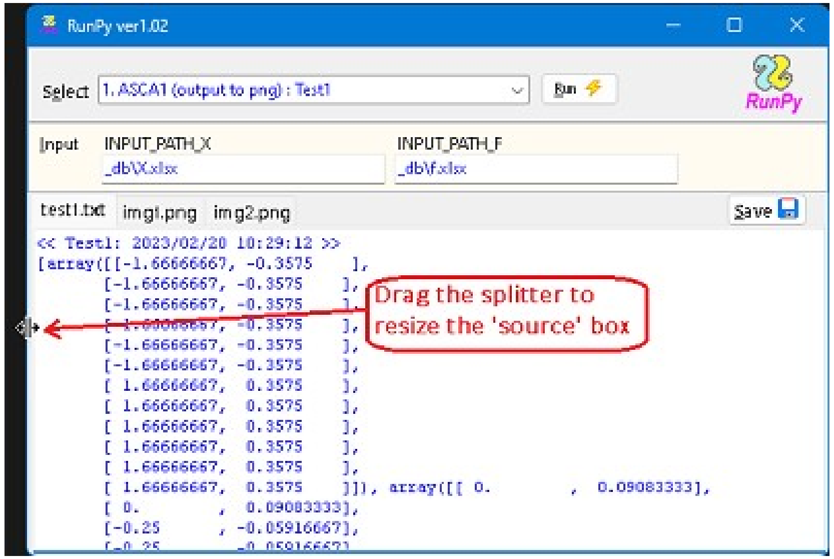
ii. If your python script create and display graph/chart, you need to import tkinter or similar GUI module;

iii. If your python script create a graph/file that need to be displayed or save in Runpy, the name of the file must be specified in the "OP:" secction of the apps in the configuration file (runpy.cfg).

RunPy - Frequently Asked Qustion (for end users)

6. Can I view the source code of the python application selected?

You can view the source code in the source box if it the source .py file exists in _db folder. By default, the source box is not display on the screen (box width=0) but you may use the mouse to drag the splitter on the left edge to open it/change its width. You can also modify and run the edited source (by clicking the Run button), and save it (by pressing ctrl+s).
If you have selected an item which is encryted in the RunPy.db database, the source code would not be displayed in the source box.

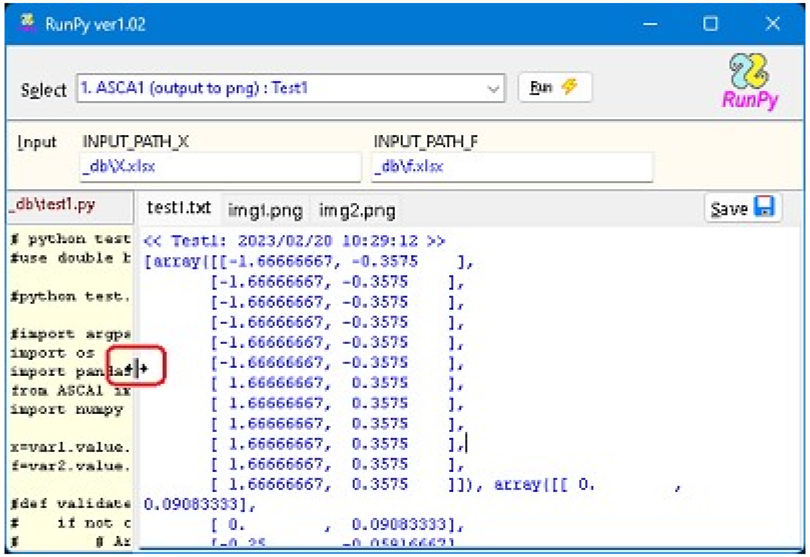


Select: 1. ASCA1 (output to png) : Test1

Input: INPUT_PATH_X _db\X.xlsx INPUT_PATH_F _db\f.xlsx

```
<< Test1: 2023/02/20 10:29:12 >>
[array([[-1.66666667, -0.3575    ],
        [-1.66666667, -0.3575    ],
        [-1.66666667, -0.3575    ],
        [-1.66666667, -0.3575    ],
        [-1.66666667, -0.3575    ],
        [-1.66666667, -0.3575    ],
        [ 1.66666667,  0.3575    ],
        [ 1.66666667,  0.3575    ],
        [ 1.66666667,  0.3575    ],
        [ 1.66666667,  0.3575    ],
        [ 1.66666667,  0.3575    ],
        [ 1.66666667,  0.3575    ]]), array([[ 0.        ,  0.09083333],
        [ 0.        ,  0.09083333],
        [-0.25      , -0.05916667],
        [-0.25      , -0.05916667]
```

Drag the splitter to resize the 'source' box



Select: 1. ASCA1 (output to png) : Test1

Input: INPUT_PATH_X _db\X.xlsx INPUT_PATH_F _db\f.xlsx

```
# python test
#use double b

#python test.

#import argpa
import os
import pandas
from ASCA1 im
import numpy

x=var1.value.
f=var2.value.

#def validate
#     if not c
#           # Ar
```

```
<< Test1: 2023/02/20 10:29:12 >>
[array([[-1.66666667, -0.3575    ],
        [-1.66666667, -0.3575    ],
        [-1.66666667, -0.3575    ],
        [-1.66666667, -0.3575    ],
        [-1.66666667, -0.3575    ],
        [-1.66666667, -0.3575    ],
        [ 1.66666667,  0.3575    ],
        [ 1.66666667,  0.3575    ],
        [ 1.66666667,  0.3575    ],
        [ 1.66666667,  0.3575    ],
        [ 1.66666667,  0.3575    ],
        [ 1.66666667,  0.3575    ]]), array([[ 0.        ,
0.09083333],
        [ 0.        ,  0.09083333],
        [-0.25      , -0.05916667]
```

RunPy - Frequently Asked Qustion (for Administrator)

I. Can I protect my python file so that the file would not be tampered?

Yes, you can do this by encrypting the source code with password in 'admin mode'. Please read FAQ below for the detailed steps. In addition, you can also encrypt the configuration file if you wish.
Runpy program always read the script/configuration file (unencrypted) from the _db folder FIRST. If it cannot find the file in _db folder, it will then try to read/ run the file from _db\runPy.db (i.e. the encrypted database).

II. How to enter into Admin mode?

i. Start RunPy program;

ii. Press Alt+/ (alternate slash) to activate password widget;

iii. Blind type (i.e. typing on key board as usual but the characters would not be displayed on the screen):

Type Slash ("/") followed by the admin password; e.g. if your password is mypw123, type /mypw123.

iv. The slash key clears everything in the buffer i.e. if you have typed a wrong character, just press / key and retype your password again.

If the password you entered is correct, an ‘Encrypt’ button will appear beside the “Run” button.

To cancel/exit admin mode, press Alt+/ (alternate slash) again.

III. How to encrypt a python file in Runpy program ?

i Test and make sure the original/unencrypted file works in RunPy;

ii In admin mode, click the "Encrypt" button;

iii Key in the name of file source file (or configuration file) and then press OK.

Note that Runpy will only encrypt 'runpy.cfg' (RunPy configuration file) or .py (python script) file.

IV. How to change admin password?

i Click on to the output box (or press alt+o);

ii Press Ctil+Alt+Backspace key to display the 'Change Password' windows;

iii Key in the current admin password;

iv Key in the new password (password must not include slash /);

v Re-key in the new password for verification;

vi Click OK button.

Check the return message to confirm that password has been changed. Sometime the change may not be effected e.g. if the the current password is not correct.