## You

xlwings add-in has a limitation that .py file containing UDF should be in the same directory as excel file to which UDF is linked. Other limitation is that the file name of .py file should be same as .xlsx file.

Is there any way to overcome these limitations? Can you search for worksround of both the above limitations?

## Copilot

Yes, there are ways to overcome these limitations of xlwings add-in:

1. **UDF in the same directory as Excel file**: You can overcome this limitation by specifying the path to your Python module in the PYTHONPATH field in the xlwings ribbon [1]. If your module is not within the same directory as the Excel file, you can point to it via the PYTHONPATH field [1]. For example, if the module was in a folder under C:\\py\_folder, you would enter C:\\py\_folder in the PYTHONPATH field [1].
2. **.py file name should be same as .xlsx file**: You can overcome this limitation by specifying the module name in the UDF\_MODULES field in the xlwings ribbon [1]. The UDF\_MODULES field only accepts the module name [2]. So, you can specify any module name regardless of the Excel file name [2]. For example, if you have a module called my\_udf.py, you would enter my\_udf in the UDF\_MODULES field [1].

Remember to click on Import Python UDFs in the xlwings tab to pick up the changes made to your Python module [1].

I hope this helps! Let me know if you have any other questions. 😊

Source: Conversation with Bing, 14/4/2024

1. [User Defined Functions (UDFs) - xlwings Documentation](https://docs.xlwings.org/en/latest/udfs.html)
2. [Can I specify an absolute location for UDF\_Modules within xlwings?](https://stackoverflow.com/questions/52975054/can-i-specify-an-absolute-location-for-udf-modules-within-xlwings)
3. [xlwings - how to import existing UDFs to a new sheet](https://stackoverflow.com/questions/49238746/xlwings-how-to-import-existing-udfs-to-a-new-sheet)
4. [unable to import UDF's via xlwings · Issue #2124 - GitHub](https://github.com/xlwings/xlwings/issues/2124)
5. [Add-in — xlwings dev documentation](https://docs.xlwings.org/en/0.12.0/addin.html)
6. [xlwings: Unable to import functions (UDFs) via xlwings ribbon](https://stackoverflow.com/questions/63665276/xlwings-unable-to-import-functions-udfs-via-xlwings-ribbon)
7. [How to improve the speed of xlwings UDFs in Excel?](https://stackoverflow.com/questions/44203521/how-to-improve-the-speed-of-xlwings-udfs-in-excel)
8. [OneDrive and SharePoint - xlwings Documentation](https://docs.xlwings.org/en/stable/onedrive_sharepoint.html)

## You

I have following UDF which takes 2 columns and several rows as array (1st column days and 2nd column has amounts):

@xw.func
@xw.arg('days\_amounts', ndim=2)
def aging\_buckets\_parallel(days\_amounts):
client = Client(n\_workers=6, threads\_per\_worker=4)
df = pd.DataFrame(days\_amounts, columns=['days', 'amount'])
df['days'] = pd.to\_numeric(df['days'], errors='coerce')
df['amount'] = pd.to\_numeric(df['amount'], errors='coerce')
ddf = dd.from\_pandas(df, npartitions=4)
aging\_list = [
(lambda day, amount: int(amount) < 0, "Advance"),
(lambda day, amount: 0 <= int(day) < 31, "Not Due"),
(lambda day, amount: 31 <= int(day) < 61, "31-60 Days"),
(lambda day, amount: 61 <= int(day) < 91, "61-90 Days"),
(lambda day, amount: 91 <= int(day) < 181, "91-180 Days"),
(lambda day, amount: 181 <= int(day) < 366, "6 Months to 1 year"),
(lambda day, amount: 366 <= int(day), "More than 1 year")
]
ddf['age\_bucket'] = ddf.map\_partitions(lambda df: df.apply(lambda row: next((label for condition, label in aging\_list if condition(row['days'], row['amount'])), None), axis=1), meta=('days', 'object'))
result\_df = ddf.compute()
result = [[item] for item in result\_df['age\_bucket'].values]
client.close()
return result

Can you modify the above so that it takes input as single array (excel range of dates) and returns like this:

If the month is within April To June returns Q1,
If the month is within July to Sep returns Q2,
If the month is within Oct to Dec returns Q3,
If the month is within Jan to Mar returns Q4.

## Copilot

Sure, I can help you modify the UDF to take a single array of dates and return the corresponding quarter based on the month of each date. Here's how you can do it:

@xw.func

@xw.arg('dates', ndim=2)

def quarter\_buckets(dates):

client = Client(n\_workers=6, threads\_per\_worker=4)

df = pd.DataFrame(dates, columns=['date'])

df['date'] = pd.to\_datetime(df['date'], errors='coerce')

quarter\_list = [

(lambda date: 4 <= date.month <= 6, "Q1"),

(lambda date: 7 <= date.month <= 9, "Q2"),

(lambda date: 10 <= date.month <= 12, "Q3"),

(lambda date: 1 <= date.month <= 3, "Q4")

]

df['quarter'] = df.apply(lambda row: next((label for condition, label in quarter\_list if condition(row['date'])), None), axis=1)

result\_df = df.compute()

result = [[item] for item in result\_df['quarter'].values]

client.close()

return result

This function will take an array of dates as input and return an array of the corresponding quarters. The quarters are defined as follows:

* **Q1**: April to June
* **Q2**: July to September
* **Q3**: October to December
* **Q4**: January to March

Please let me know if you have any other questions! 😊

## You

When loading UDF I get the error in spite of dask having been already installed:

import dask.dataframe as dd
from dask.distributed import Client

Error:

D:\Programs\Python\Lib\site-packages\dask\dataframe\_*init*\_.py:31: FutureWarning:
Dask dataframe query planning is disabled because dask-expr is not installed.

You can install it with pip install dask[dataframe] or conda install dask.
This will raise in a future version.

warnings.warn(msg, FutureWarning)
pythoncom error: Python error invoking COM method.

Traceback (most recent call last):
File "D:\Programs\Python\Lib\site-packages\win32com\server\policy.py", line 305, in *Invoke*
return self.*invoke*(dispid, lcid, wFlags, args)
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "D:\Programs\Python\Lib\site-packages\win32com\server\policy.py", line 310, in *invoke*
return S\_OK, -1, self.*invokeex*(dispid, lcid, wFlags, args, None, None)
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "D:\Programs\Python\Lib\site-packages\win32com\server\policy.py", line 639, in *invokeex*
return func(\*args)
^^^^^^^^^^^
File "D:\Programs\Python\Lib\site-packages\xlwings\com\_server.py", line 231, in Call
return ToVariant(getattr(obj, method)(\*pargs, \*\*kwargs))
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "D:\Programs\Python\Lib\site-packages\xlwings\udfs.py", line 703, in import\_udfs
module = get\_udf\_module(module\_name, xl\_workbook)
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "D:\Programs\Python\Lib\site-packages\xlwings\udfs.py", line 381, in get\_udf\_module
module = import\_module(module\_name)
^^^^^^^^^^^^^^^^^^^^^^^^^^
File "D:\Programs\Python\Lib\importlib\_*init*\_.py", line 90, in import\_module
return \_bootstrap.\_gcd\_import(name[level:], package, level)
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "", line 1387, in \_gcd\_import
File "", line 1360, in \_find\_and\_load
File "", line 1331, in \_find\_and\_load\_unlocked
File "", line 935, in \_load\_unlocked
File "", line 995, in exec\_module
File "", line 488, in *call\_with\_frames\_removed
File "D:\OneDrive - 0yt2k\excel examples including add-in xlwings and connected python - vba - dlls, etc\xlwings\_trial.py", line 6, in
import dask.dataframe as dd
File "D:\Programs\Python\Lib\site-packages\dask\dataframe\_init*.py", line 110, in
raise ImportError(msg) from e
ImportError: Dask dataframe requirements are not installed.

Please either conda or pip install as follows:

conda install dask # either conda install
python -m pip install "dask[dataframe]" --upgrade # or python -m pip install

## Copilot

I'm sorry to hear that you're having trouble with Dask and xlwings. The error message suggests that Dask's dataframe requirements are not installed. Even though you have Dask installed, it seems like the specific requirements for Dask's dataframe functionality are missing.

You can try installing the Dask dataframe requirements using either of the following commands:

conda install dask

or

python -m pip install "dask[dataframe]" --upgrade

These commands should install the necessary requirements for Dask's dataframe functionality [1]. After running these commands, try importing Dask and loading your UDF again.

If the problem persists, it might be helpful to check the configuration of xlwings according to its Config Hierarchy [3].

Please let me know if this helps or if you need further assistance! 😊

Source: Conversation with Bing, 14/4/2024

1. [Dask[dataframe] "requirements are not installed" on AWS lambda](https://stackoverflow.com/questions/72396425/daskdataframe-requirements-are-not-installed-on-aws-lambda)
2. [Troubleshooting - xlwings Documentation](https://docs.xlwings.org/en/latest/troubleshooting.html)
3. [Dask [dataframe] "requirements are not installed" on AWS lambda](https://github.com/dask/dask/issues/9132)

## You

I did as above The error now is:

## Copilot

I'm sorry, but it seems like the error message was not included in your last message. Could you please provide the error message that you're seeing now? This will help me better understand the issue and provide a more accurate solution. 😊

## You

Imported functions from the following modules: xlwings\_trial
pythoncom error: Python error invoking COM method.

Traceback (most recent call last):
File "D:\Programs\Python\Lib\site-packages\win32com\server\policy.py", line 305, in *Invoke*
return self.*invoke*(dispid, lcid, wFlags, args)
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "D:\Programs\Python\Lib\site-packages\win32com\server\policy.py", line 310, in *invoke*
return S\_OK, -1, self.*invokeex*(dispid, lcid, wFlags, args, None, None)
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "D:\Programs\Python\Lib\site-packages\win32com\server\policy.py", line 639, in *invokeex*
return func(\*args)
^^^^^^^^^^^
File "D:\Programs\Python\Lib\site-packages\xlwings\com\_server.py", line 235, in CallUDF
res = call\_udf(script, fname, args, this\_workbook, FromVariant(caller))
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "D:\Programs\Python\Lib\site-packages\xlwings\udfs.py", line 502, in call\_udf
ret = func(\*args)
^^^^^^^^^^^
File "D:\OneDrive - 0yt2k\excel examples including add-in xlwings and connected python - vba - dlls, etc\xlwings\_trial.py", line 203, in quarter\_buckets
result\_df = df.compute()
^^^^^^^^^^
File "D:\Programs\Python\Lib\site-packages\pandas\core\generic.py", line 6299, in **getattr**
return object.**getattribute**(self, name)
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
AttributeError: 'DataFrame' object has no attribute 'compute'. Did you mean: 'compare'?

## Copilot

The error message you're seeing is due to the fact that the compute() method is not available for a Pandas DataFrame. The compute() method is a feature of Dask DataFrame [2].

In your code, it seems like df is a Pandas DataFrame, not a Dask DataFrame. That's why you're seeing the error AttributeError: 'DataFrame' object has no attribute 'compute'.

If you want to use Dask's compute() method, you should convert your Pandas DataFrame to a Dask DataFrame. Here's how you can do it:

import dask.dataframe as dd

# Convert the Pandas DataFrame to a Dask DataFrame

ddf = dd.from\_pandas(df, npartitions=4)

# Now you can use the compute() method

result\_df = ddf.compute()

Please try this and let me know if it helps or if you have any other questions! 😊

Source: Conversation with Bing, 14/4/2024

1. [AttributeError: 'DataFrame' object has no attribute 'compute'](https://stackoverflow.com/questions/54898315/attributeerror-dataframe-object-has-no-attribute-compute)
2. [AttributeError: module 'xlwings' has no attribute 'load'](https://stackoverflow.com/questions/67020577/attributeerror-module-xlwings-has-no-attribute-load)
3. [How to resolve AttributeError: 'DataFrame' object has no attribute](https://stackoverflow.com/questions/38134643/how-to-resolve-attributeerror-dataframe-object-has-no-attribute)
4. [pandas - AttributeError 'dataframe' object has no attribute](https://stackoverflow.com/questions/51502263/pandas-attributeerror-dataframe-object-has-no-attribute)
5. <https://docs.xlwings.org/en/stable/api.html>
6. <https://github.com/xlwings/xlwings/issues/1555>
7. <https://geektechstuff.com/2020/06/13/attributeerror-partially-initialized-module-has-no-attribute-python/>

## You

Can you modify below based of=n dask dataframe:

@xw.func
@xw.arg('dates', ndim=2)
def quarter\_buckets(dates):
client = Client(n\_workers=6, threads\_per\_worker=4)
df = pd.DataFrame(dates, columns=['date'])
ddf = dd.from\_pandas(df, npartitions=4)
ddf['date'] = ddf.to\_datetime(ddf['date'], errors='coerce')
quarter\_list = [
(lambda date: 4 <= date.month <= 6, "Q1"),
(lambda date: 7 <= date.month <= 9, "Q2"),
(lambda date: 10 <= date.month <= 12, "Q3"),
(lambda date: 1 <= date.month <= 3, "Q4")
]
ddf['quarter'] = ddf.apply(lambda row: next((label for condition, label in quarter\_list if condition(row['date'])), None), axis=1)
result\_df = ddf.compute()
result = [[item] for item in result\_df['quarter'].values]
client.close()
return result

## Copilot

Sure, I can help you modify the UDF to work with a Dask DataFrame. However, please note that Dask's apply() function doesn't work exactly the same way as Pandas' apply(). In Dask, apply() doesn't inherently support row-wise operations, which is what you're trying to do here.

But don't worry, we can still achieve the desired functionality by using Dask's map\_partitions() function, which applies a function to each partition of the DataFrame. Here's how you can modify your function:

@xw.func

@xw.arg('dates', ndim=2)

def quarter\_buckets(dates):

client = Client(n\_workers=6, threads\_per\_worker=4)

df = pd.DataFrame(dates, columns=['date'])

ddf = dd.from\_pandas(df, npartitions=4)

ddf['date'] = dd.to\_datetime(ddf['date'], errors='coerce')

quarter\_list = [

(lambda date: 4 <= date.month <= 6, "Q1"),

(lambda date: 7 <= date.month <= 9, "Q2"),

(lambda date: 10 <= date.month <= 12, "Q3"),

(lambda date: 1 <= date.month <= 3, "Q4")

]

ddf['quarter'] = ddf.map\_partitions(lambda df: df['date'].apply(lambda date: next((label for condition, label in quarter\_list if condition(date)), None)), meta=('date', 'object'))

result\_df = ddf.compute()

result = [[item] for item in result\_df['quarter'].values]

client.close()

return result

In this modified function, I've replaced the apply() function with map\_partitions(). This will apply the lambda function to each partition of the DataFrame. The meta argument is used to specify the output format.

Please let me know if this helps or if you have any other questions! 😊

## You

Forget about dask. USe only pandas. data is not so big to use dask. It slows down the operation of UDF.

## Copilot

Sure, we can modify the function to use only Pandas. Here's how you can do it:

import pandas as pd

@xw.func

@xw.arg('dates', ndim=2)

def quarter\_buckets(dates):

df = pd.DataFrame(dates, columns=['date'])

df['date'] = pd.to\_datetime(df['date'], errors='coerce')

quarter\_list = [

(lambda date: 4 <= date.month <= 6, "Q1"),

(lambda date: 7 <= date.month <= 9, "Q2"),

(lambda date: 10 <= date.month <= 12, "Q3"),

(lambda date: 1 <= date.month <= 3, "Q4")

]

df['quarter'] = df['date'].apply(lambda date: next((label for condition, label in quarter\_list if condition(date)), None))

result = [[item] for item in df['quarter'].values]

return result

In this modified function, I've removed the Dask-related code and used only Pandas. The apply() function is used to apply the lambda function to the 'date' column of the DataFrame.

Please let me know if this helps or if you have any other questions! 😊

## You

To call above function from a different excel file in different path I defined .py file with path in PYTHONPATH variable on xlwings tab as: D:\OneDrive - 0yt2k\Excel Examples Including add-in xlwings and connected python - vba - dlls, etc\xlwings\_trial.py. I then inserted udf name as quarter\_buckets in udf module in xlwings as stated by you above.

I get error:

xlwings server running, clsid={17184AA2-2126-4E0B-B552-F5EB2D6E87FC}, event loop on 13208
pythoncom error: Python error invoking COM method.

Traceback (most recent call last):
File "D:\Programs\Python\Lib\site-packages\win32com\server\policy.py", line 305, in *Invoke*
return self.*invoke*(dispid, lcid, wFlags, args)
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "D:\Programs\Python\Lib\site-packages\win32com\server\policy.py", line 310, in *invoke*
return S\_OK, -1, self.*invokeex*(dispid, lcid, wFlags, args, None, None)
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "D:\Programs\Python\Lib\site-packages\win32com\server\policy.py", line 639, in *invokeex*
return func(\*args)
^^^^^^^^^^^
File "D:\Programs\Python\Lib\site-packages\xlwings\com\_server.py", line 231, in Call
return ToVariant(getattr(obj, method)(\*pargs, \*\*kwargs))
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "D:\Programs\Python\Lib\site-packages\xlwings\udfs.py", line 703, in import\_udfs
module = get\_udf\_module(module\_name, xl\_workbook)
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "D:\Programs\Python\Lib\site-packages\xlwings\udfs.py", line 381, in get\_udf\_module
module = import\_module(module\_name)
^^^^^^^^^^^^^^^^^^^^^^^^^^
File "D:\Programs\Python\Lib\importlib\_*init*\_.py", line 90, in import\_module
return \_bootstrap.\_gcd\_import(name[level:], package, level)
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "", line 1387, in \_gcd\_import
File "", line 1360, in \_find\_and\_load
File "", line 1324, in \_find\_and\_load\_unlocked
ModuleNotFoundError: No module named 'quarter\_buckets'

## Copilot

I'm sorry to hear that you're having trouble with xlwings. The error message ModuleNotFoundError: No module named 'quarter\_buckets' suggests that Python is unable to find a module named 'quarter\_buckets'.

In the xlwings ribbon, the PYTHONPATH should be the directory where your Python file is located, and the UDF\_MODULES should be the name of your Python file without the .py extension [1].

From your message, it seems like you have set the PYTHONPATH to the full path of your Python file, including the .py extension. This should be the directory only. And in UDF\_MODULES, you have entered the function name 'quarter\_buckets' instead of the module name.

Here's how you can correct it:

1. Set PYTHONPATH to D:\OneDrive - 0yt2k\Excel Examples Including add-in xlwings and connected python - vba - dlls, etc\
2. Set UDF\_MODULES to xlwings\_trial

After making these changes, try importing the UDFs again.

If the problem persists, it might be helpful to check the configuration of xlwings according to its Config Hierarchy [3].

Please let me know if this helps or if you need further assistance! 😊

Source: Conversation with Bing, 14/4/2024

1. [xlwings: Unable to import functions (UDFs) via xlwings ribbon](https://stackoverflow.com/questions/63665276/xlwings-unable-to-import-functions-udfs-via-xlwings-ribbon)
2. [ModuelNotFound Error when importing function -used to work ... - GitHub](https://github.com/xlwings/xlwings/issues/2079)
3. [ModuleNotFoundError: No module named 'xlwings' - Stack Overflow](https://stackoverflow.com/questions/72939166/modulenotfounderror-no-module-named-xlwings)
4. [Fix "ModuleNotFoundError: No module named 'xlwings'" error](https://modulenotfound.com/xlwings.html)
5. <https://docs.xlwings.org/en/stable/udfs.html>

## You

See the error after above change:

pythoncom error: Python error invoking COM method.

Traceback (most recent call last):
File "D:\Programs\Python\Lib\site-packages\win32com\server\policy.py", line 305, in *Invoke*
return self.*invoke*(dispid, lcid, wFlags, args)
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "D:\Programs\Python\Lib\site-packages\win32com\server\policy.py", line 310, in *invoke*
return S\_OK, -1, self.*invokeex*(dispid, lcid, wFlags, args, None, None)
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "D:\Programs\Python\Lib\site-packages\win32com\server\policy.py", line 639, in *invokeex*
return func(\*args)
^^^^^^^^^^^
File "D:\Programs\Python\Lib\site-packages\xlwings\com\_server.py", line 231, in Call
return ToVariant(getattr(obj, method)(\*pargs, \*\*kwargs))
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "D:\Programs\Python\Lib\site-packages\xlwings\udfs.py", line 703, in import\_udfs
module = get\_udf\_module(module\_name, xl\_workbook)
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "D:\Programs\Python\Lib\site-packages\xlwings\udfs.py", line 381, in get\_udf\_module
module = import\_module(module\_name)
^^^^^^^^^^^^^^^^^^^^^^^^^^
File "D:\Programs\Python\Lib\importlib\_*init*\_.py", line 90, in import\_module
return \_bootstrap.\_gcd\_import(name[level:], package, level)
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "", line 1387, in \_gcd\_import
File "", line 1360, in \_find\_and\_load
File "", line 1324, in \_find\_and\_load\_unlocked
ModuleNotFoundError: No module named 'Workings-new'

## Copilot

I'm sorry to hear that you're still having trouble. The error message ModuleNotFoundError: No module named 'Workings-new' suggests that Python is unable to find a module named 'Workings-new'.

In the xlwings ribbon, the PYTHONPATH should be the directory where your Python file is located, and the UDF\_MODULES should be the name of your Python file without the .py extension [1].

From your message, it seems like you have set the UDF\_MODULES to 'Workings-new'. If 'Workings-new' is the name of your Python file, make sure that it does not include any spaces or hyphens, as these are not allowed in Python module names [4].

You can rename your Python file to a valid module name (for example, workings\_new.py), and then set UDF\_MODULES to the new name (in this case, workings\_new).

After making these changes, try importing the UDFs again.

If the problem persists, it might be helpful to check the configuration of xlwings according to its Config Hierarchy [3].

Please let me know if this helps or if you need further assistance! 😊

Source: Conversation with Bing, 14/4/2024

1. [ModuleNotFoundError: No module named 'xlwings' - Stack Overflow](https://stackoverflow.com/questions/72939166/modulenotfounderror-no-module-named-xlwings)
2. [ModuelNotFound Error when importing function -used to work ... - GitHub](https://github.com/xlwings/xlwings/issues/2079)
3. [Fix "ModuleNotFoundError: No module named 'xlwings'" error](https://modulenotfound.com/xlwings.html)
4. [xlwings: Unable to import functions (UDFs) via xlwings ribbon](https://stackoverflow.com/questions/63665276/xlwings-unable-to-import-functions-udfs-via-xlwings-ribbon)
5. <https://docs.xlwings.org/en/stable/udfs.html>

## You

Why my python path as: D:\OneDrive - 0yt2k\Excel Examples Including add-in xlwings and connected python - vba - dlls, etc and file named xlwings\_trial.py in visual UDF modules box on xlwings tabs are not recognized? I get following error: (FYI my current excel file is workings\_new.xlsm but I have already defined as above still not recognized in xlwings?)

pythoncom error: Python error invoking COM method.

Traceback (most recent call last):
File "D:\Programs\Python\Lib\site-packages\win32com\server\policy.py", line 305, in *Invoke*
return self.*invoke*(dispid, lcid, wFlags, args)
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "D:\Programs\Python\Lib\site-packages\win32com\server\policy.py", line 310, in *invoke*
return S\_OK, -1, self.*invokeex*(dispid, lcid, wFlags, args, None, None)
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "D:\Programs\Python\Lib\site-packages\win32com\server\policy.py", line 639, in *invokeex*
return func(\*args)
^^^^^^^^^^^
File "D:\Programs\Python\Lib\site-packages\xlwings\com\_server.py", line 231, in Call
return ToVariant(getattr(obj, method)(\*pargs, \*\*kwargs))
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "D:\Programs\Python\Lib\site-packages\xlwings\udfs.py", line 703, in import\_udfs
module = get\_udf\_module(module\_name, xl\_workbook)
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "D:\Programs\Python\Lib\site-packages\xlwings\udfs.py", line 381, in get\_udf\_module
module = import\_module(module\_name)
^^^^^^^^^^^^^^^^^^^^^^^^^^
File "D:\Programs\Python\Lib\importlib\_*init*\_.py", line 90, in import\_module
return \_bootstrap.\_gcd\_import(name[level:], package, level)
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "", line 1387, in \_gcd\_import
File "", line 1360, in \_find\_and\_load
File "", line 1324, in \_find\_and\_load\_unlocked
ModuleNotFoundError: No module named 'Workings-new'

## Copilot

I'm sorry to hear that you're still having trouble. The error message ModuleNotFoundError: No module named 'Workings-new' suggests that Python is unable to find a module named 'Workings-new'.

In the xlwings ribbon, the PYTHONPATH should be the directory where your Python file is located, and the UDF\_MODULES should be the name of your Python file without the .py extension [1].

From your message, it seems like you have set the UDF\_MODULES to 'Workings-new'. If 'Workings-new' is the name of your Python file, make sure that it does not include any spaces or hyphens, as these are not allowed in Python module names [4].

You can rename your Python file to a valid module name (for example, workings\_new.py), and then set UDF\_MODULES to the new name (in this case, workings\_new).

After making these changes, try importing the UDFs again.

If the problem persists, it might be helpful to check the configuration of xlwings according to its Config Hierarchy [3].

Please let me know if this helps or if you need further assistance! 😊

Source: Conversation with Bing, 14/4/2024

1. [ModuleNotFoundError: No module named 'xlwings' - Stack Overflow](https://stackoverflow.com/questions/72939166/modulenotfounderror-no-module-named-xlwings)
2. [ModuelNotFound Error when importing function -used to work ... - GitHub](https://github.com/xlwings/xlwings/issues/2079)
3. [Fix "ModuleNotFoundError: No module named 'xlwings'" error](https://modulenotfound.com/xlwings.html)
4. [xlwings: Unable to import functions (UDFs) via xlwings ribbon](https://stackoverflow.com/questions/63665276/xlwings-unable-to-import-functions-udfs-via-xlwings-ribbon)
5. <https://docs.xlwings.org/en/stable/udfs.html>

## You

Finally I could make that work. But now the UDF which you gave me above needs to be changed to have different logic:

If Date is between 01/04 to 15/06 - output: "01/04 To 15/06"
If Date is between 16/06 to 15/09 - output: "16/06 To 15/09"
If Date is between 16/09 to 15/12 - output: "16/09 To 15/12"
If Date is between 16/12 to 15/03 - output: "16/12 To 15/03"
If date is between 16/03 to 31/03 - output: "16/03 To 31/03"

## Copilot

Sure, I can help you modify the UDF to implement the new logic. Here's how you can do it:

import pandas as pd

@xw.func

@xw.arg('dates', ndim=2)

def date\_buckets(dates):

df = pd.DataFrame(dates, columns=['date'])

df['date'] = pd.to\_datetime(df['date'], errors='coerce')

date\_list = [

(lambda date: (date.month == 4 and date.day >= 1) or (date.month == 5) or (date.month == 6 and date.day <= 15), "01/04 To 15/06"),

(lambda date: (date.month == 6 and date.day >= 16) or (date.month == 7) or (date.month == 8) or (date.month == 9 and date.day <= 15), "16/06 To 15/09"),

(lambda date: (date.month == 9 and date.day >= 16) or (date.month == 10) or (date.month == 11) or (date.month == 12 and date.day <= 15), "16/09 To 15/12"),

(lambda date: (date.month == 12 and date.day >= 16) or (date.month == 1) or (date.month == 2) or (date.month == 3 and date.day <= 15), "16/12 To 15/03"),

(lambda date: (date.month == 3 and date.day >= 16 and date.day <= 31), "16/03 To 31/03")

]

df['bucket'] = df['date'].apply(lambda date: next((label for condition, label in date\_list if condition(date)), None))

result = [[item] for item in df['bucket'].values]

return result

In this modified function, I've updated the lambda functions in date\_list to check the date ranges according to your new logic.

Please let me know if this helps or if you have any other questions! 😊