

ISF INTEGRATION OF MACRO BOTS

BOT REQUIREMENT

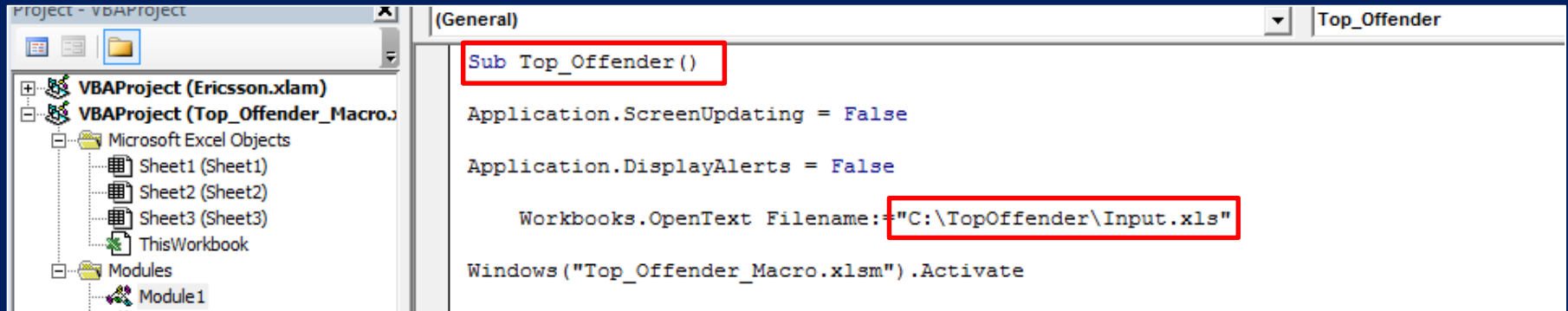
REQUIRED MACRO CHANGES

- Macro file name should not contain any Blank spaces and Special characters (Can use underscore).
- Your macro code should be written on Module level, it should not be on sheet level.
- If your code is on sheet level then do either of any one changes in your code as below:
 - Either create a new module and call your sheet code from new module.
 - Or shift your Sheet level code in Module.

REQUIRED MACRO CHANGES FOR INPUT

- Need to remove all hardcode Input Paths from your code. Examples as below:

- Old code:



```

Sub Top_Offender()

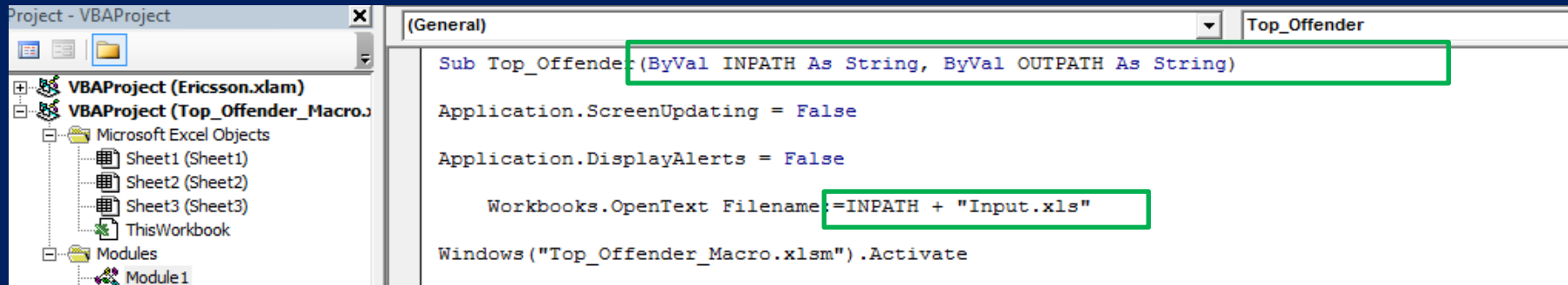
    Application.ScreenUpdating = False

    Application.DisplayAlerts = False

    Workbooks.OpenText Filename:="C:\TopOffender\Input.xls"

    Windows("Top_Offender_Macro.xlsm").Activate
  
```

- New code:



```

Sub Top_Offender(ByVal INPATH As String, ByVal OUTPATH As String)

    Application.ScreenUpdating = False

    Application.DisplayAlerts = False

    Workbooks.OpenText Filename:=INPATH + "Input.xls"

    Windows("Top_Offender_Macro.xlsm").Activate
  
```

- As shown in above images, please changes your hard coded path accordingly. And make changes in your complete code to remove hard coded path for input path.

REQUIRED MACRO CHANGES FOR OUTPUT

- You must save macro output in OUTPATH variable to remove all hardcode Output Paths as shown in below code:

- Old code:

```
ActiveWorkbook.SaveAs Filename:="C:\TopOffender\Output Final.xlsx", _  
    FileFormat:=xlOpenXMLWorkbook, CreateBackup:=False  
Sheets("Normalized Score").Select
```

- New code:

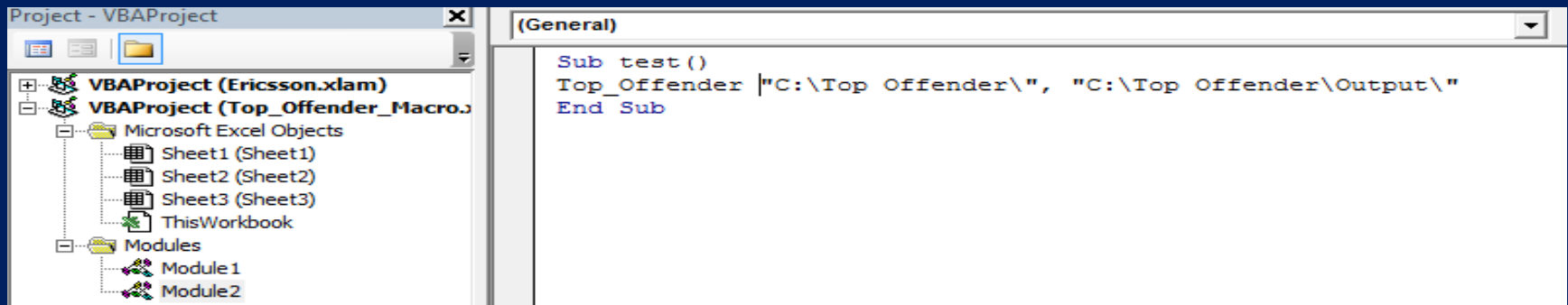
```
ActiveWorkbook.SaveAs Filename:=OUTPATH + "Output Final.xlsx", _  
    FileFormat:=xlOpenXMLWorkbook, CreateBackup:=False  
Sheets("Normalized Score").Select
```

- If you are closing your macro file programmatically at last step of macro then comment or remove it as in below example:

```
'Workbooks("Top_Offender_Macro.xlsm").Close SaveChanges:=False
```

UNIT TESTING OF MACRO AFTER CHANGES

- Before sharing your macro code to ISF-RPA team, do unit testing of your macro for desired results. For unit testing use below steps:
 - Step1- Create a new module.
 - Step2- Create a new method in this module and call your main method from this this new method.
 - Step3- Pass two parameters when calling your main method.
 - Step4- After successful macro execution, please check your desired output in OUTPATH folder.



ISF INTEGRATION OF PYTHON BOTS

BOT REQUIREMENT

REQUIRED PYTHON CHANGES

- Your python code should not contain any UI related code, if any it needs to be removed.
- The python code should have main method accepting 2 parameters as below snippet-

```
def doProcess(INPATH,OUTPATH):  
    '''  
    Logic Part  
    '''  
if __name__ == '__main__':  
    if len(sys.argv) < 3:  
        print("Please provide input/output path")  
        sys.exit()  
    INPATH=sys.argv[1]  
    OUTPATH=sys.argv[2]  
    doProcess(INPATH,OUTPATH)
```


REQUIRED PYTHON CHANGES FOR INPUT

- Need to remove all hardcode Input Paths from your code. Examples as below:

- Old code:

```
def NTX():  
    wb = load_workbook('C:/Users/eshukch/workspace/NTX-AUDIT-PYTHON/INPATH/Audit_Database.xlsx')
```

- New code:

```
def NTX(INPATH,OUTPATH):  
    wb = load_workbook(INPATH+'Audit_Database.xlsx')
```

- As shown in above images, please changes your hard coded path accordingly. And make changes in your complete code to remove hard coded path for input path.

REQUIRED PYTHON CHANGES FOR OUTPUT

- You must save Python output in OUTPATH variable to remove all hardcode Output Paths as shown in below code:

- Old code:

```
directory = "C:/Users/eshukch/workspace/NTX-AUDIT-PYTHON/INPATH/Reports"
if not os.path.isdir(directory):
    os.mkdir(directory)
report_filename = os.path.join(directory, "OUTPUT_Final.xlsx")
CR_filename = os.path.join(directory, CR_filename)
wb1.save(filename = report_filename)
```

- New code:

```
directory = OUTPATH+'Reports'
if not os.path.isdir(directory):
    os.mkdir(directory)
report_filename = os.path.join(directory, "OUTPUT_Final.xlsx")
CR_filename = os.path.join(directory, CR_filename)
wb1.save(filename = report_filename)
```

- Convert the .py file into EXE.



ERICSSON