

Global Solutions from the World's Largest Data-Focused CRO

SM04: To VBA or not to VBA? That is the question.

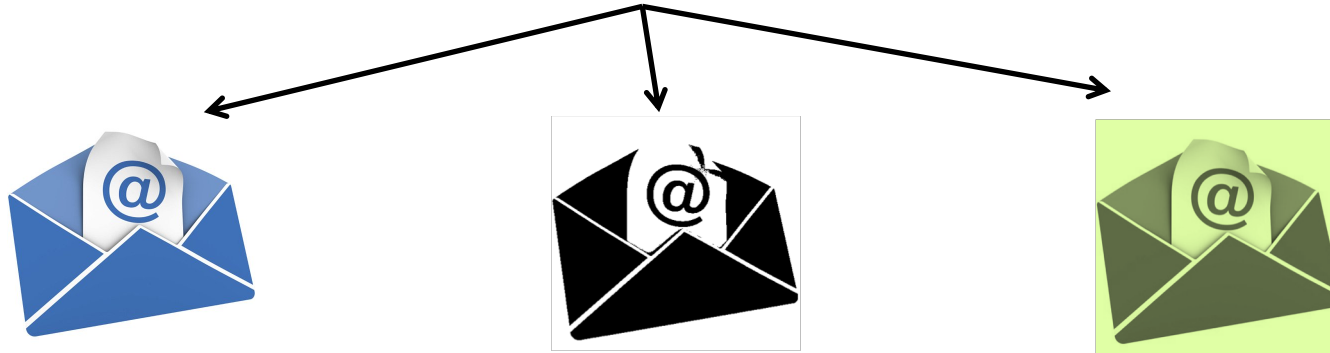
Piotr Karasiewicz

Agenda

- Introduction
- Sending emails using SAS
- Sending emails using Visual Basic for Applications
 - Includes introduction to Object Oriented Programming
- SAS and VBA combined together
- Summary

Task ahead

Name	Email	Subject of Email	Custom Text	Attachment
Name1 Surname1	Name1.Surname1@quanticate.com	Subject Line 1		...\PHUSE Frankfurt\attachment1.txt
Name2 Surname2	Name2.Surname2@quanticate.com	Subject Line 2	Custom Text 2	...\PHUSE Frankfurt\attachmen2.txt
Name3 Surname3	Name3.Surname3@quanticate.com	Subject Line 3		...\PHUSE Frankfurt\attachment1.txt



SAS - supported email protocols

- **MAPI** - Messaging Application Program Interface – default – it uses default email client installed on PC, e.g. Outlook or Mozilla Thunderbird.
- **SMTP** - Simple Mail Transfer Protocol – this method bypasses windows Client and connects to email server directly.
- **VIM** - Vendor Independent Mail - VIM can be used in combination with Lotus Notes or cc:Mail.

SAS options

OPTIONS

```
emailsys=MAPI  
emailid="Name.Surname";
```

OPTIONS

```
emailsys=SMTP  
emailhost="smtp.server.local"  
emailport=25  
emailid="name@quanticate.com"  
emailpw="{SAS002}...";
```

```
PROC PWENCODE  
  in="sample password";  
RUN;
```



SAS – sending single email

```
FILENAME mailbox EMAIL to=("name1@quanticate.com")
                        cc=("name2@quanticate.com")
                        subject="PHUSE SM04"
                        attach=("path1\filename1");

DATA _null_;
  FILE mailbox;
  PUT 'Hi Piotr,';
  PUT 'Files for your analysis are attached. Enjoy!';
  PUT 'Regards,';

RUN;

FILENAME mailbox clear;
```

SAS – single email with directives

```
DATA _null_;  
  FILE mailbox;  
  PUT "!em_newmsg!";  
  PUT "!em_to! " "name1@quanticate.com";  
  PUT "!em_cc! " "name2@quanticate.com";  
  PUT "!em_subject!" "PHUSE SM04";  
  PUT "Hi Piotr,";  
  PUT "Custom Text";  
  PUT "Regards,";  
  PUT "!em_attach!" "path1\filename1";  
  
RUN;
```

SAS email directives

!em_XXX!, where XXX is one of the following: newmsg, to, subject, cc, bcc, attach, importance, replyto, abort & send

SAS – bulk emails

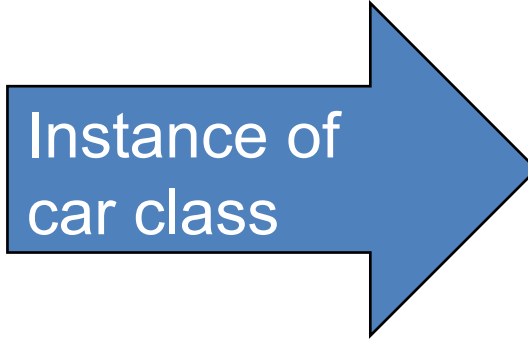
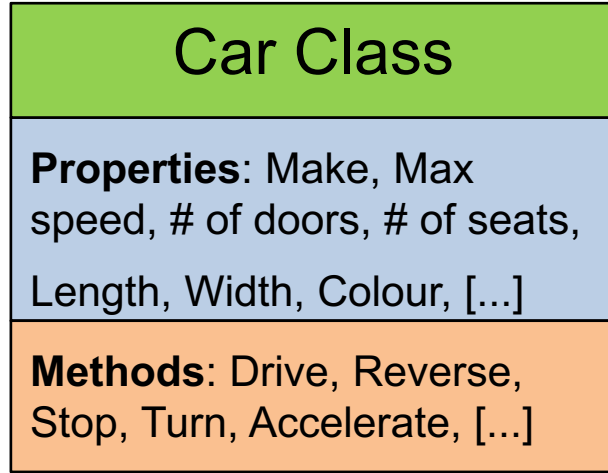
```
DATA _null_;  
  SET email_list END=EOF;  
  FILE mailbox;  
  PUT "!em_newmsg!";  
  PUT "!em_to! " email;  
  PUT "!em_subject!" subject;  
  PUT "email body";  
  PUT "!em_send!";  
  _DELAY=SLEEP(60,1);  
  IF EOF THEN "!em_abort!";  
RUN;
```

Variables from “email_list” dataset
which contains all details on recipients

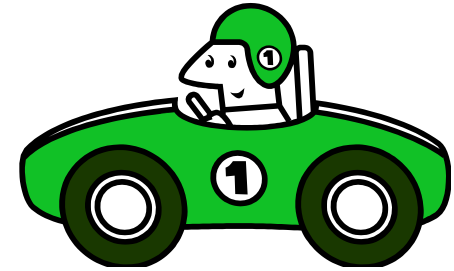
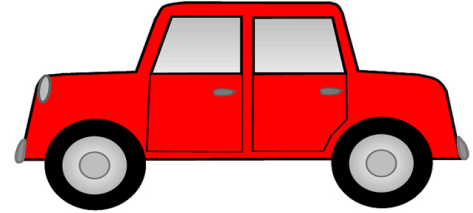
If EMAILSYS=SMTP this line reduces the risk
of email landing in SPAM

This prevents the last recipient from receiving
two identical emails

Object Oriented Programming (OOP)



Car Objects



OOP – illustration in Python

Define class

```
class phuse_presentation(object):
```

```
    # Properties
```

```
    def __init__(self, topic, duration, presenter):
```

```
        self.topic=topic
```

```
        self.duration=duration
```

```
        self.presenter=presenter
```

```
    # Sample Method
```

```
    def dointro(self):
```

```
        print ("topic: " + self.topic, "\n", "presenter: " + self.presenter)
```

```
        print ("duration: " + str(self.duration) + " minutes")
```

Instantiate object, add properties and use dointro method

```
sm04 = phuse_presentation("To VBA or not to VBA..",20,"Piotr Karasiewicz")
```

```
sm04.dointro()
```

VB Script (*.vbs) vs. VBA

- VBScript is simply “lightweight” subset of VBA
- **Definition of variables/objects:** in VBA it is very common to use “Dim var1 as String” notation. In VBScript correct syntax is: “Dim var1”
- **Definition of arrays:** in VBA it is quite common to write: “Dim array1(1 to 5, 1 to 10) as Integer – 5x10 array created. In VBScript: “Dim array1(4, 9)” – here the index starts from 0, rather than 1 which SAS programmers may be used to.

VBA – Mailitem class - extract

Properties	Property Type	Methods
Attachments	Object	Close
Body	String	Copy
CC	String	Display
HTMLBody	String	Forward
Recipients	Object	PrintOut
RTFBody	Variant	Reply
Sender	String	ReplyAll
To	String	Send

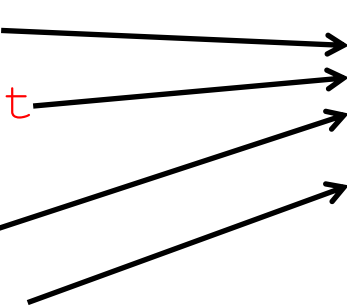
Sample VBA code

```
Sub email1()  
    Dim OutApp As Object  
    Dim OutMail As Object  
    Set OutApp = CreateObject("Outlook.Application")  
    Set OutMail = OutApp.CreateItem(olMailItem)  
    Dim email_attachment As String  
    email_attachment = "path\filename"  
    With OutMail  
        .Attachments.Add (email_attachment)  
    End With  
End Sub
```

VBA – sending single email

With OutMail

```
.To = programmer_email  
.Subject = email_subject  
.BodyFormat = 2  
.HTMLBody = HTMLBody  
.Attachments.Add (email_att)  
.Display  
.Send
```



Variables defined
earlier in the code:

End With

VBA – sending bulk emails

'Loading excel file into array emails()

'Looping through elements of array

For i = 0 To number_of_recipients-1

 'define key variables (i.e. programmer_email, etc.)
 based on emails[i]

 ' [code for single email]

Next

SAS + VBA

```
OPTIONS noxwait noxsync;  
%LET _file=path\xlsm\file\with\VBA\macro\file.xlsm;  
X "start excel /r &_file";  
FILENAME fileref1 DDE "Excel|system";  
DATA _null_;  
  FILE fileref1;  
  PUT ' [RUN("Macro1")] ';  
RUN;  
FILENAME fileref1 clear;
```

Summary

- All three methods fits the purpose
- It is difficult to point at the superior method

Questions



Contact:

Name : Piotr Karasiewicz

Company : Quanticate International Ltd.

Email : Piotr.Karasiewicz@Quanticate.com