

Automating SAS Integration Technologies via Excel

Tim Walters

InfoTech Marketing

Two Different Contexts

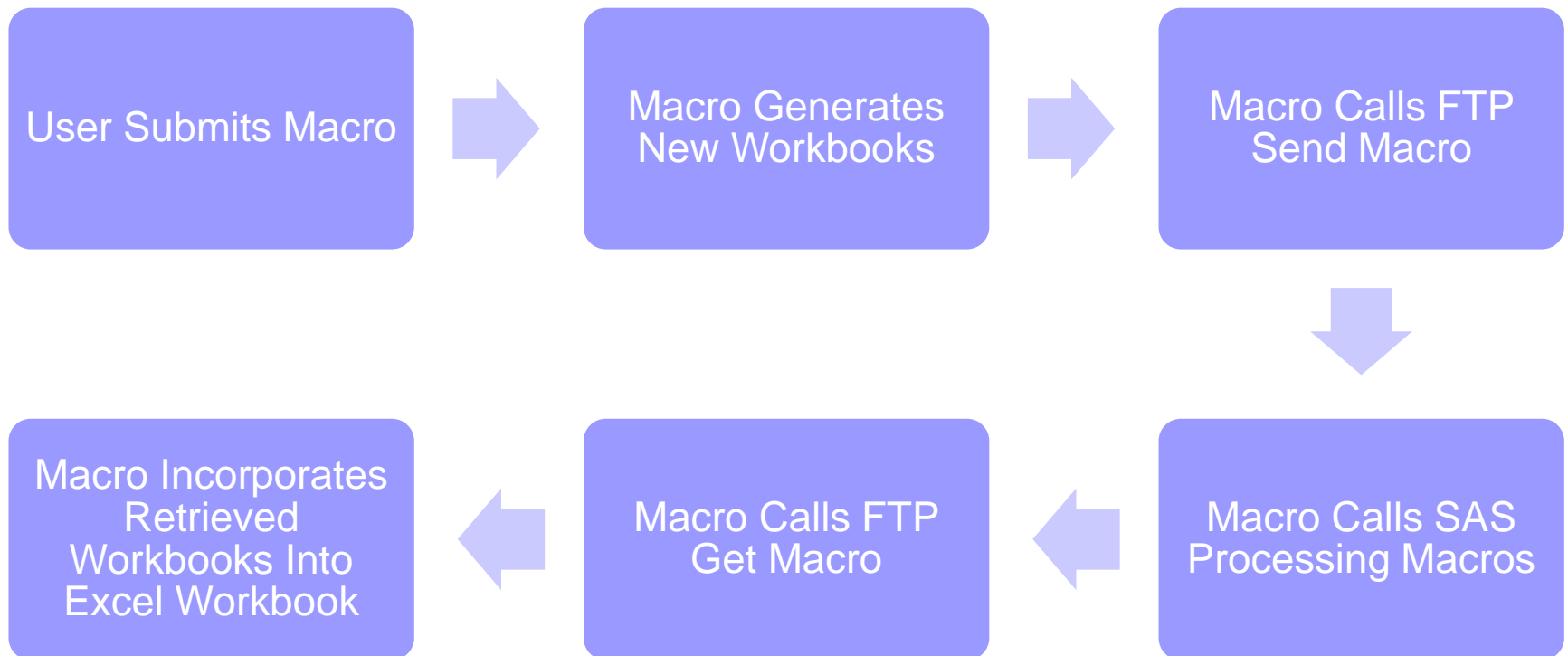
- On-Demand -- Someone using Excel who runs a macro, produces Excel workbooks, FTPs to SAS server, runs SAS, FTPs from SAS server, incorporates SAS output into Excel
- Scheduled – Uses Windows Scheduler to open Excel workbook, FTPs to SAS server, runs SAS, FTPs from SAS server

ON DEMAND

Requirements

- User must have copy of SAS E.G. on their client machine. Workbooks can be on networked drive, but user must have own copy of SAS E.G.
- All folders and files FTP'd must not have spaces in their names. For example, they must be like
v:\Forecasting_Models\CurrrentModel.xls
m and not v:\Forecasting Models\Current Model.xlsm.

System Overview



Excel Macro Overview – Located in Module 1

Sub DM_Analysis()

Application.ScreenUpdating=False

Application.DisplayAlerts=False

Application.EnableEvents=False

curr_path = ThisWorkbook.Path

curr_book = ThisWorkbook.Name

' DIRECT MAIL MODULE

Workbooks.Add

new_book = ActiveWorkbook.Name

Workbooks(curr_book).Worksheets("DM Curves").Activate

ActiveWorkbook.Sheets("DM Curves").Copy _

after:=Workbooks(new_book).Sheets("Sheet1")

fname = curr_path & "\DM_Curves.csv"

Workbooks(new_book).Worksheets("DM Curves").SaveAs _

Filename:=fname, FileFormat:=xlCSV

ActiveWorkbook.Close

[More File Creations]

Call PutCSVFile

Call runme

Call GetCSVFile

'paste dm forecast into current model workbook

Call pasteforecast

end

1. User Submits Macro & 2. Macro Generates New Workbooks

- User Submits DM_Analysis Macro via Excel Developer or Keyboard

Sub DM_Analysis()

Application.ScreenUpdating=False

Application.DisplayAlerts=False

Application.EnableEvents=False

curr_path = ThisWorkbook.Path

curr_book = ThisWorkbook.Name

' DIRECT MAIL MODULE

Workbooks.Add

new_book = ActiveWorkbook.Name

Workbooks(curr_book).Worksheets("DM Curves").Activate

ActiveWorkbook.Sheets("DM Curves").Copy _

after:=Workbooks(new_book).Sheets("Sheet1")

fname = curr_path & "\DM_Curves.csv"

Workbooks(new_book).Worksheets("DM Curves").SaveAs _

Filename:=fname, FileFormat:=xlCSV

ActiveWorkbook.Close

[More File Creations]

Call PutCSVFile

3. Macro Calls FTP Send Macro (Part 1)

```
Sub PutCSVFile()
```

```
Dim strDirectoryList As String
```

```
Dim IStr_Dir As String
```

```
Dim lInt_FreeFile01 As Integer
```

```
Dim lInt_FreeFile02 As Integer
```

```
On Error GoTo Err_Handler
```

```
IStr_Dir = "h:\" 'Directory to save FTP transmittal files
```

```
lInt_FreeFile01 = FreeFile
```

```
lInt_FreeFile02 = FreeFile
```

```
strDirectoryList = IStr_Dir & "\Directory"
```

```
' Delete previous completion file if not deleted
```

```
If Dir(strDirectoryList & ".out") <> "" Then Kill (strDirectoryList & ".out")
```


3. Macro Calls FTP Send Macro (Part 2)

" Create text file with FTP commands

Open strDirectoryList & ".txt" For Output As #lInt_FreeFile01

Print #lInt_FreeFile01, "open 10.128.138.28" 'SAS server IP Address

Print #lInt_FreeFile01, "twalters" 'My username on the SAS server

Print #lInt_FreeFile01, "xxxxx" 'My password on the SAS server

Print #lInt_FreeFile01, "cd /vg04/twalters" 'Directory to put files on SAS server

Print #lInt_FreeFile01, "binary"

Print #lInt_FreeFile01, "send " & ThisWorkbook.Path & "\Direct_Mail_Dates_Master.txt
Direct_Mail_Dates_Master.txt"

Print #lInt_FreeFile01, "send " & ThisWorkbook.Path & "\DM_Curves.csv
DM_Curves.csv"

[More Files]

Print #lInt_FreeFile01, "bye"

Close #lInt_FreeFile01

3. Macro Calls FTP Send Macro (Part 3)

' Create Batch program

Open strDirectoryList & ".bat" For Output As #lnt_FreeFile02

Print #lnt_FreeFile02, "ftp -s:" & strDirectoryList & ".txt"

Print #lnt_FreeFile02, "Echo ""Complete"" > " & strDirectoryList & ".out"

Close #lnt_FreeFile02

" Invoke Directory List generator

Shell (strDirectoryList & ".bat"), vbHide ", vbMinimizedNoFocus

'Wait for completion

Do While Dir(strDirectoryList & ".out") = ""

 DoEvents

Loop

Application.Wait (Now + TimeValue("0:00:03"))

3. Macro Calls FTP Send Macro (Part 4)

```
' Clean up files
```

```
If Dir(strDirectoryList & ".bat") <> "" Then Kill (strDirectoryList & ".bat")
```

```
If Dir(strDirectoryList & ".out") <> "" Then Kill (strDirectoryList & ".out")
```

```
If Dir(strDirectoryList & ".txt") <> "" Then Kill (strDirectoryList & ".txt")
```

```
bye:
```

```
Exit Sub
```

```
Err_Handler:
```

```
MsgBox "Error : " & Err.Number & vbCrLf & "Description : " & Err.Description,  
vbCritical
```

```
Resume bye
```

```
End Sub
```

4. Macro Calls SAS Processing Macros (Part 1)

```
Sub runme()
```

```
    Dim app          'The EG 4.3 application
```

```
    Call dowork
```

```
End Sub
```

```
Sub dowork()
```

```
    Dim prjObject    'The project object for EG
```

```
    Dim codeObj      'The code object for the project
```

```
    Dim prjName      'The Project we are making
```

```
    Dim codeFile     'The name of the file that contains SAS code
```

```
    Dim logFile      'The path to the log file that will be saved
```

```
    Dim codeServer   'Active Server to run SAS code
```

```
    '-----
```

```
    ' Set the file names. (These could be parameters.)
```

```
    '-----
```

```
    prjName = ThisWorkbook.Path & "\\Test_Forecasting_Project.egp" 'Project Name
```

```
    codeFile = ThisWorkbook.Path & "\\DM_Forecast_v4.sas"         'SAS Code File
```

```
    logFile = ThisWorkbook.Path & "\\DM_Forecast_Log_File.log"   'Log File
```

```
    codeServer = "SASApp"    'SAS Code Server
```



4. Macro Calls SAS Processing Macros (Part 2)

' Set up the 4.3 project and open it.

```
Set app = CreateObject("SASEGObjectModel.Application.4.3")
```

```
If Checkerror("CreateObject") = True Then
```

```
    Stop
```

```
    Exit Sub
```

```
End If
```

```
Set prjObject = app.New()
```

```
If Checkerror("app.New") = True Then
```

```
    Stop
```

```
    Exit Sub
```

```
End If
```

' open the file and put the text into the fileString variable

```
Dim fileSys, fReadStream, fileString
```

```
Set fileSys = CreateObject("Scripting.FileSystemObject")
```

```
Set fReadStream = fileSys.OpenTextFile(codeFile, 1, False)
```

```
fileString = fReadStream.ReadAll()
```

4. Macro Calls SAS Processing Macros (Part 3)

```
' add the new code object to the project
  Set codeObj = prjObject.CodeCollection.Add

' Set the server and text for the code
  codeObj.Server = codeServer
  codeObj.Text = fileString
  codeObj.Run          ' Run the code
  codeObj.Log.SaveAs logFile ' Save the log file

' Save the project with the code
  SaveCloseProject prjObject, prjName
```

End Sub

4. Macro Calls SAS Processing Macros (Part 4)

```
Sub SaveCloseProject(prjObject, prjName)
  ' Save the project
  prjObject.SaveAs prjName

  If Checkerror("Project.Save") = True Then
    Exit Sub
  End If

  ' Close the project
  prjObject.Close
  If Checkerror("Project.Close") = True Then
    Exit Sub
  End If

End Sub
```

4. Macro Calls SAS Processing Macros (Part 5)

' Looks for an error message and displays it in a message box, but only

' if the line that starts with:

' ' MsgBox strmsg

' has had the single quote removed.

Function Checkerror(fnName)

 Checkerror = False

 Dim strmsg

 Dim errNum

 If Err.Number <> 0 Then

 strmsg = "Error #" & Hex(Err.Number) & vbCrLf & "In Function " & fnName & vbCrLf
& Err.Description

 ' MsgBox strmsg 'Uncomment this line if you want to be notified via MessageBox of
Errors in the script.

 Checkerror = True

 End If

End Function

4. Macro Calls SAS Processing Macros (Part 6)

```
data WORK.DIRECT_MAIL_DATES package_days(keep=key_pkg_cd elapsed_days  
mail_day_of_week type mail_month);  
  infile '/vg04/twalters/Direct_Mail_Dates_Master.txt' delimiter = '09'x MISSOVER DSD  
lrecl=32767 firstobs=2 ;
```

```
proc export data=member_by_month outfile="/vg04/twalters/DM_Forecast.csv"  
dbms=csv replace;  
run;
```

5. Macro Calls FTP Get Macro

Exactly the same as the FTP Send macro, except the line files change to:

```
Print #lInt_FreeFile01, "recv DM_Forecast.csv " & ThisWorkbook.Path &
"\DM_Forecast.csv "
```

```
Print #lInt_FreeFile01, "recv WB_Forecast.csv " & ThisWorkbook.Path &
"\WB_Forecast.csv "
```

```
Print #lInt_FreeFile01, "recv DM_Curves_Computed.csv " & ThisWorkbook.Path &
"\DM_Curves_Computed.csv "
```

```
Print #lInt_FreeFile01, "recv WB_Curves_Computed.csv " & ThisWorkbook.Path &
"\WB_Curves_Computed.csv "
```

6. Macro Incorporates Retrieved Workbooks Into Excel Workbook

```
Sub pasteforecast()
```

```
'paste dm forecast into current model workbook
```

```
  Sheets("DM_Forecast").Select
```

```
  Cells.Select
```

```
  Selection.Clear
```

```
  fname = curr_path & "\DM_Forecast.csv"
```

```
  Workbooks.Open Filename:=fname
```

```
  Cells.Select
```

```
  Selection.Copy
```

```
  Workbooks(curr_book).Worksheets("DM_Forecast").Activate
```

```
  Range("A1").Select
```

```
  ActiveSheet.Paste
```

```
  Workbooks("DM_Forecast.csv").Close
```

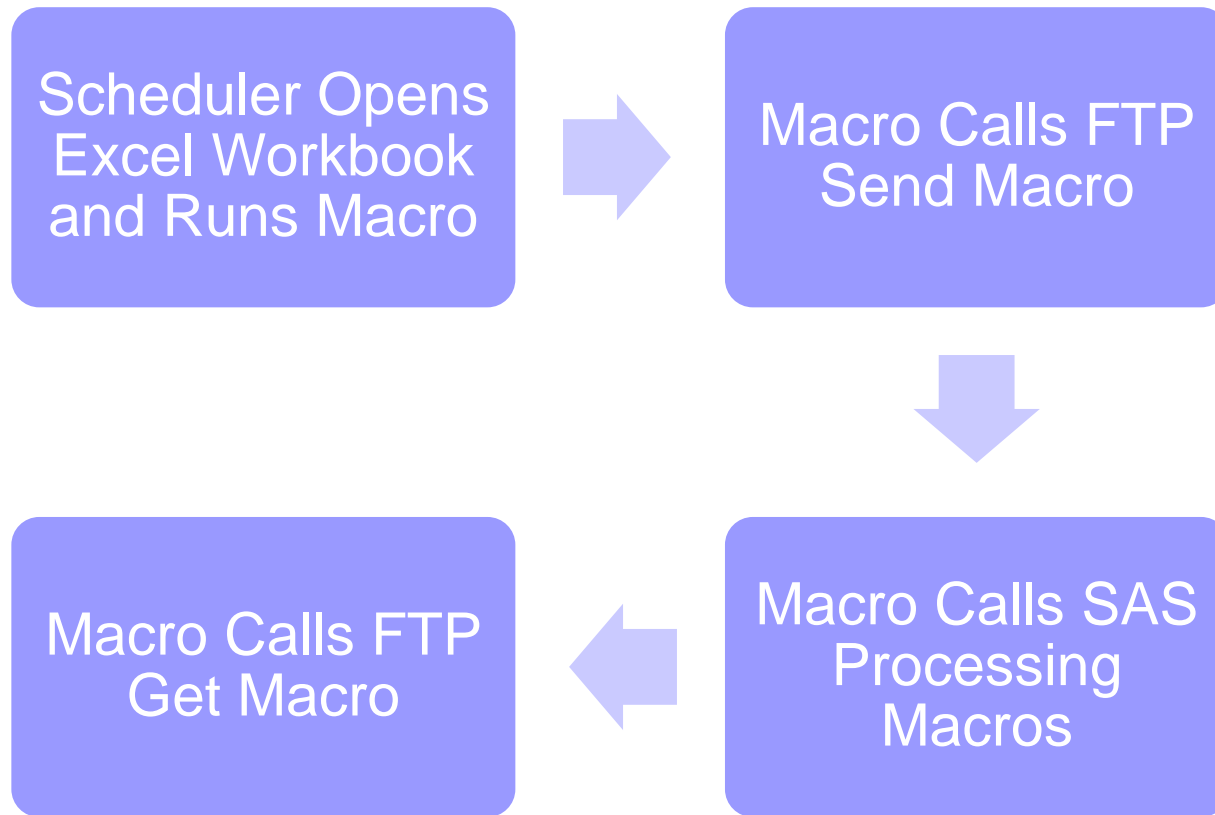
```
End sub
```

SCHEDULED

Requirements

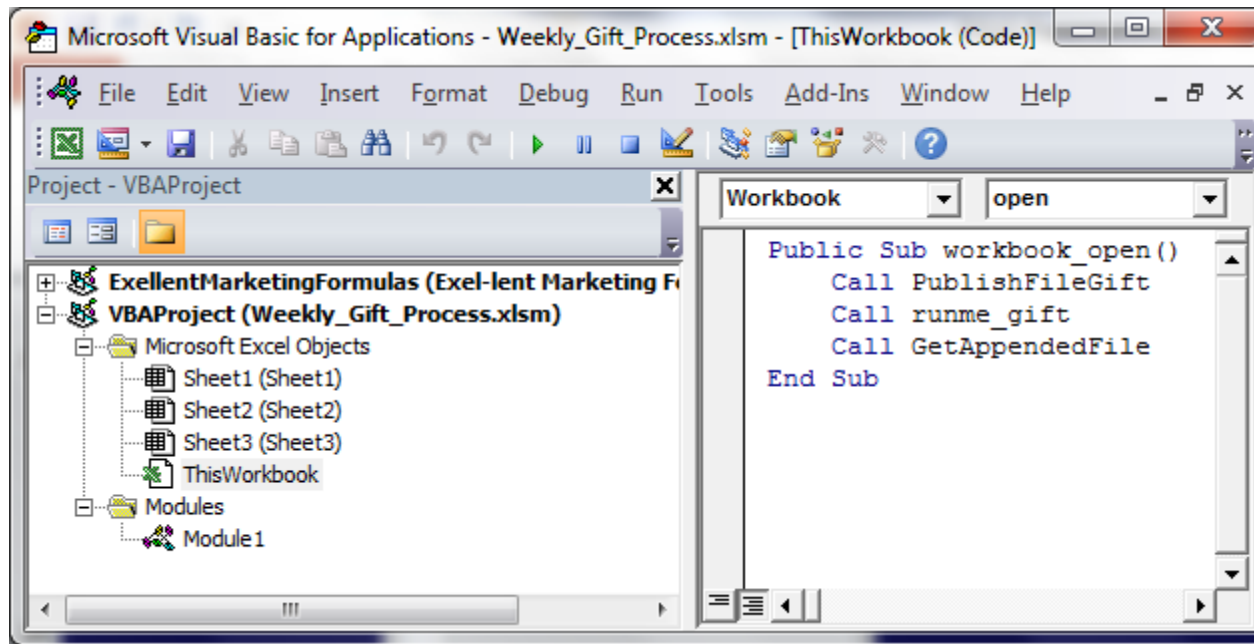
- User must have copy of SAS E.G. on their client machine. Workbooks can be on networked drive, but user must have own copy of SAS E.G.
- All folders and files FTP'd must not have spaces in their names. For example, they must be like v:\Forecasting_Models\CurrentModel.xlsm and not v:\Forecasting Models\Current Model.xlsm.
- Close all Excel workbooks.

System Overview



1. Scheduler Opens Excel Workbook and Runs Macro

- Scheduler Line
V:\IVS\BusSol&Ana\Weekly_Gift_Process\Weekly_Gift_Process.xlsm
- VBA – must be in ThisWorkbook and use the Workbook_Open event



2. Macro Calls FTP Send Macro

- Basically the same as before
- Code below shows how to use folder with spaces in the name, e.g.
V:\FDN\FDN_Public\Gift Planning\FdnFulFiles\FNDFULL040812.txt

```
curr_file = "FDNFULL" & Format(Date - 1, "mmddy") & ".txt"
```

'have to copy file to h drive because the original location has a space in the folder name

```
FileCopy "V:\FDN\FDN_Public\Gift Planning\FdnFulFiles\" & curr_file, "h:\" & _  
curr_file
```

```
Kill (IStr_Dir & curr_file)
```


3. Macro Calls SAS Processing Macros

- Exactly the same as before, just different programs and files
prjName = ThisWorkbook.Path & "\Automated_Gift_Project.egp" 'Project Name
codeFile = ThisWorkbook.Path & "\Process_Weekly_Gift_v3.sas" 'SAS Code File
logFile = ThisWorkbook.Path & "\Process_Weekly_Gift_Log_File.log" 'Log File

4. Macro Calls FTP Send Macro

- Basically the same as before
- Code below shows how to use folder with spaces in the name, e.g.
V:\FDN\FDN_Public\Gift Planning\FdnFulFiles\FNDFULL040812.txt

```
curr_file = "FDNFULL" & Format(Date - 1, "mmddy") & "_Appended.txt"
```

'have to copy file from h drive because the original location has a space in the folder name

```
FileCopy "h:\" & curr_file, "V:\FDN\FDN_Public\Gift Planning\FdnFulFiles\" & _  
curr_file
```

```
Kill (IStr_Dir & curr_file)
```

For More Information

Tim Walters

InfoTech Marketing

720-732-4588

tim@infotechmarketing.net