Paper 102-27

A Better Desktop Than Windows: Using Excel to Organize, View, Launch and Document SAS® Programs

Ted Conway, Ted Conway Consulting, Inc., Chicago, IL

ABSTRACT

With the help of a few simple macros provided in this paper, Excel can come in handy for any SAS user that's trying to organize, view, launch, and even document large numbers of Windows-based SAS programs that are run on a regular basis.

INTRODUCTION

Maybe you've inherited one too many SAS programs from others.

Or perhaps you're just too darn prolific yourself!

In any event, you've managed to accumulate all those supposedly "one-shot" SAS programs that you now find yourself running on a regular basis.

Daily.

Weekly.

Monthly.

Quarterly.

Annually.

On Request.

Admit it.

You're starting to feel just a bit overwhelmed.

Quite frankly, it's getting nearly impossible to keep things straight yourself, let alone trying to explain it to others.

Let's take a look at your options. You could:

- Check out monster.com for a quick way out.
- Put in for a transfer and dump things on some other poor sap.
- Keep scheduling your time-off around these tasks.
- Try to organize things so the processing can be easily grasped and run by anyone - even consultants and bosses!

While the first two options are indeed tempting at times, for now let's go ahead and explore the last one.

What could you possibly do to better organize things?

You're undoubtedly already familiar with the old time-tested method of using dog-eared, coffee-stained, marked-up, yellowing sheets of paper (in fact, you're probably using it right now!).

Then again, you could join those who've tried their very best to use Windows groups and sub-groups and sub-sub-groups and…well, you get the picture.

Perhaps you've even seen desperate attempts to instill order with the naming conventions from hell.

No doubt you've thought that there has to be a better way.

THE SOLUTION

Wouldn't it be nice if there was an easy way to:

- Organize all of your SAS programs in one place?
- View them with a simple click of a mouse?
- Run them with a simple double-click of a mouse?
- Annotate them with text and/or links to external documents or web pages?

There is - and the software you need is probably already installed on your workstation.

Excel!

THE DETAILS

Yes, that Excel!

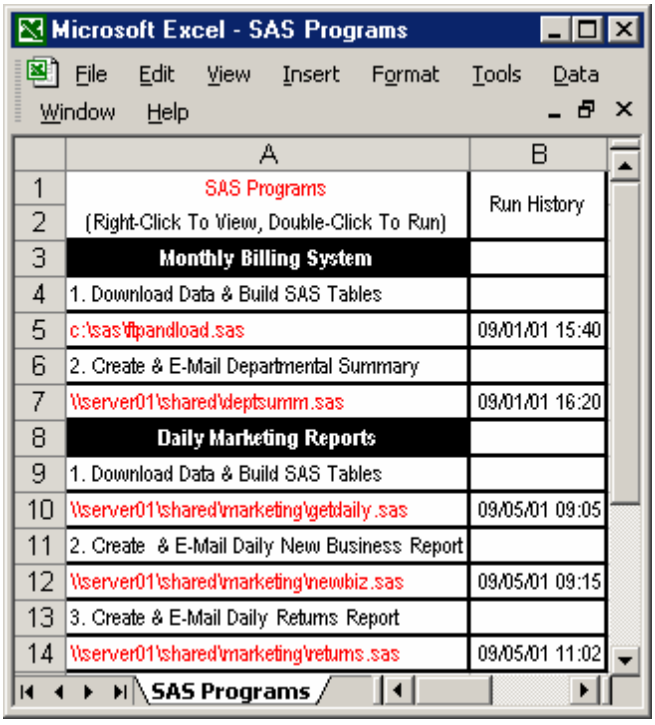
If nothing else, you'll have to admit that bosses and consultants have certainly demonstrated that Excel is great for making lists.

Well, by adding a few simple macros, those Excel lists can be turned into something that's really useful for you!

In our case, we're going to turn a list of SAS programs housed in an Excel worksheet into a "living", self-documenting run guide.

The following example provides a simple illustration of how you can use Excel to compile an inventory of SAS programs for a number of systems.

It rather tersely conveys a wealth of useful information, including run frequencies, the order in which programs are to be run, program descriptions, source code file names, and run-time history. For some systems, this alone may be a step up as far as documentation goes.



	A	B
1	SAS Programs	Run History
2	(Right-Click To View, Double-Click To Run)	
3	**Monthly Billing System**	
4	1. Download Data & Build SAS Tables	
5	c:\sas\ftpandload.sas	09/01/01 15:40
6	2. Create & E-Mail Departmental Summary	
7	\\server01\shared\deptsumm.sas	09/01/01 16:20
8	**Daily Marketing Reports**	
9	1. Download Data & Build SAS Tables	
10	\\server01\shared\marketing\getdaily.sas	09/05/01 09:05
11	2. Create & E-Mail Daily New Business Report	
12	\\server01\shared\marketing\newbiz.sas	09/05/01 09:15
13	3. Create & E-Mail Daily Returns Report	
14	\\server01\shared\marketing\returns.sas	09/05/01 11:02

SAS Programs

However, the addition of just a few simple macros (which are provided in the next section) is what really brings the run guide to life by providing the following features:

- Right-Clicking the mouse on a SAS program name brings up the source code in Notepad for quick review.
- Double-Clicking the mouse on a SAS program name not only executes the selected SAS program, but (optionally) also automatically inserts a cell containing the date and time that the program was last run to provide a LIFO run history.

THE CODE

You'll need the following four Excel macros to:

- Capture right-clicks (*Workbook_SheetBeforeRightClick)*
- View SAS programs with Notepad (*ViewSAS*)
- Capture double-clicks (*Workbook_SheetBeforeDoubleClick*)
- Submit SAS Programs & record DateTimeStamps (*RunSAS*)

1. CAPTURE RIGHT MOUSE-CLICKS (VIEW REQUESTS)

```
Private Sub
Workbook_SheetBeforeRightClick(ByVal Sh As
Object, ByVal Target As Excel.Range, Cancel
As Boolean)

'--> View Right-Clicked SAS Program

If UCase$(Right$(Trim(ActiveCell.Value), 4))
= ".SAS" Then
  Cancel = True
  ViewSAS
End If

End Sub
```

2. VIEW A SAS PROGRAM WITH NOTEPAD

```
Sub ViewSAS()

'--> Verify View Request (Exit If Cancelled)

a = MsgBox("View " & ActiveCell.Value & "?",
vbOKCancel, "View SAS Program Confirmation")

If a = vbCancel Then
  Exit Sub
End If

'--> View SAS Program (With Notepad)

z = Shell("notepad " & ActiveCell.Value,
vbNormalFocus)

End Sub
```

3. CAPTURE DOUBLE-CLICKS (SUBMIT REQUESTS)

```
Private Sub
Workbook_SheetBeforeDoubleClick(ByVal Sh As
Object, ByVal Target As Excel.Range, Cancel
As Boolean)

'--> Run Double-Clicked SAS Program

If UCase$(Right$(Trim(ActiveCell.Value), 4))
= ".SAS" Then
  Cancel = True
  RunSAS
End If

End Sub
```

4. SUBMIT SAS PROGRAM & RECORD DATETIMESTAMP

```
Sub RunSAS()

'--> Verify Run Request (Exit If Cancelled)

a = MsgBox("Run " & ActiveCell.Value & "?",
vbOKCancel, "Run SAS Program Confirmation")

If a = vbCancel Then
  Exit Sub
End If

'--> Run SAS Program

z = Shell("c:\sas\sas.exe -initstmt '%include
""" & ActiveCell.Value & """ / source2;
run;'", vbNormalFocus)

'--> If Requested, Insert Run Time/Date In
FIFO Order
'    ('Run History' Must Appear In First Row
Of Column To Right Of Program Name)

If UCase$(Trim(Cells(1, ActiveCell.Column +
1).Value)) = "RUN HISTORY" Then
 Cells(ActiveCell.Row, ActiveCell.Column +
1).Select
 Selection.Insert Shift:=xlToRight
 Cells(ActiveCell.Row, ActiveCell.Column +
1).Select
 Selection.Copy
 Cells(ActiveCell.Row, ActiveCell.Column -
1).Select
 Selection.PasteSpecial Paste:=xlFormats,
Operation:=xlNone, SkipBlanks:= _
     False, Transpose:=False
 Selection.Value = Date & " " & Time()
 Cells(ActiveCell.Row, ActiveCell.Column -
1).Select
 Application.CutCopyMode = False
End If
```

FURTHER POSSIBILITIES

For the purposes of this paper, things have been intentionally kept rather simple and limited to SAS applications - use your imagination and you'll find many more creative and useful ways to use the basic underlying concept.

For example, the author has used it to run other PC applications, execute commands on remote Unix/mainframe systems (and return the output to the PC), FTP files, etc.

Also, other built-in Excel features can be exploited to enhance systems documentation via cell comments or even hyperlinks to other internal or external documents or web sites. Try indentation to document hierarchical relationships amongst programs. For the ambitious, Excel & VBA's multiple selection features can even be used to allow you to run blocks of highlighted programs at once.

CONCLUSION

The portable and ubiquitous nature of Excel workbooks makes it a breeze to share truly useful run guide documentation with your coworkers on a network, via diskettes, or even e-mail.

So go ahead and finally take that vacation – they'll cover for you!

CONTACT INFORMATION

Your comments and questions are valued and encouraged.

Contact the author at tedconway@aol.com.

TRADEMARK CITATION