Paper 274-2012

SAS/FSP(r) Provides the Best Customizable Form Viewer

Peter Crawford, Crawford Software Consultancy, London, UK

ABSTRACT

Drilling through data to examine exceptions (outliers or missing) you want to see a lot of detail about one row. You need a "form viewer". SAS/FSP® software demonstrates, perhaps the best.

There are aspects to SAS/FSP that are not matched among the related products i.e. SAS/AF®, ViewTable, Web Report Studio, nor the downloadable extension for Enterprise Guide® software that promises an FSEDIT screen.

The feature least supported (in my opinion) among these alternatives is the customizable Form Viewer of FSBROWSE.

After "what" and "how", this paper will show how customizing is easily re-usable in the explorer of SAS Display Manager.

Although it may be “off-strategy” as SAS/FSP won't work in SAS Enterprise Guide nor SAS Portal, this paper shows the benefits of having a simple customizable form.

INTRODUCTION

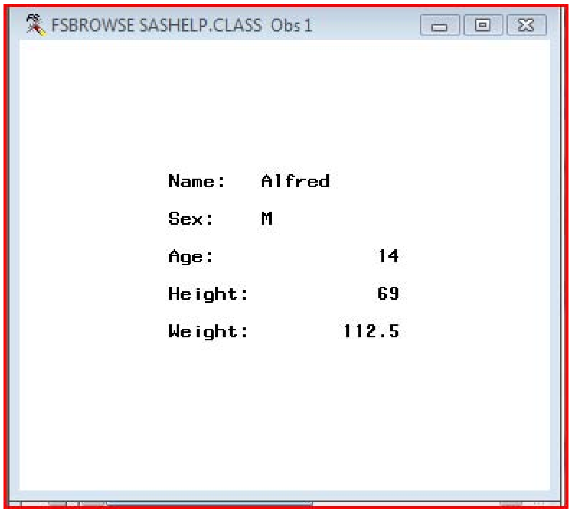
Although SAS/FSP might be unfamiliar you could find it useful for arranging multiple columns of a wide table into a convenient form. This paper will describe and demonstrate how it can work for you. For those who run SAS in batch, for example, through Enterprise Guide, I hope this paper will demonstrate the need to support and the benefit of having, reusable form customization.

After describing the facility, we will walk through usage and customizing of a form view.

WHAT IS FSBROWSE FORM VIEWER

SAS/FSP was the first product in the SAS portfolio to provide interactive viewers of SAS data. In this paper I draw almost no attention to the “update capability”. It is the ease for reviewing multiple columns I see as the big advantage.

A very simple example might look like



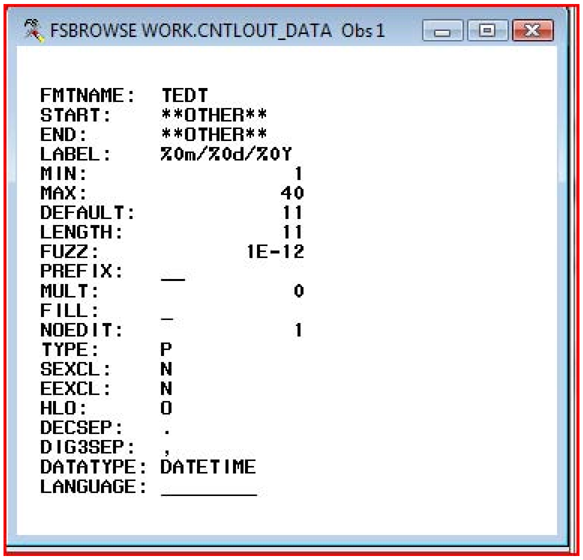
```
Name:    Alfred
Sex:     M
Age:                 14
Height:              69
Weight:           112.5
```

Figure 1. Very simple form layout

This sample data set does not demonstrate the benefits that come from customizing a form viewer. That benefit becomes more obvious when there are more columns. One easily created set of data for demonstration, is generated from PROC FORMAT, like:

```
proc format cntlout= cntlout_data ;
run ;
```

Looking at that CNTLOUT data, the form opens looking very basic:



```
FSBROWSE WORK.CNTLOUT_DATA  Obs 1

FMTNAME:  TEDT
START:    **OTHER**
END:      **OTHER**
LABEL:    %0m/%0d/%0Y
MIN:                1
MAX:               40
DEFAULT:           11
LENGTH:            11
FUZZ:           1E-12
PREFIX:   __
MULT:               0
FILL:     _
NOEDIT:             1
TYPE:     P
SEXCL:    N
EEXCL:    N
HLO:      O
DECSEP:   .
DIG3SEP:  ,
DATATYPE: DATETIME
LANGUAGE: ________
```

Figure 2. Table of more columns needing basic layout improved

The viewer can be opened with command

```
fsbrowse cntlout_data
```

or, if it is the most recently created data set, enough is just the command abbreviation

```
fsb
```

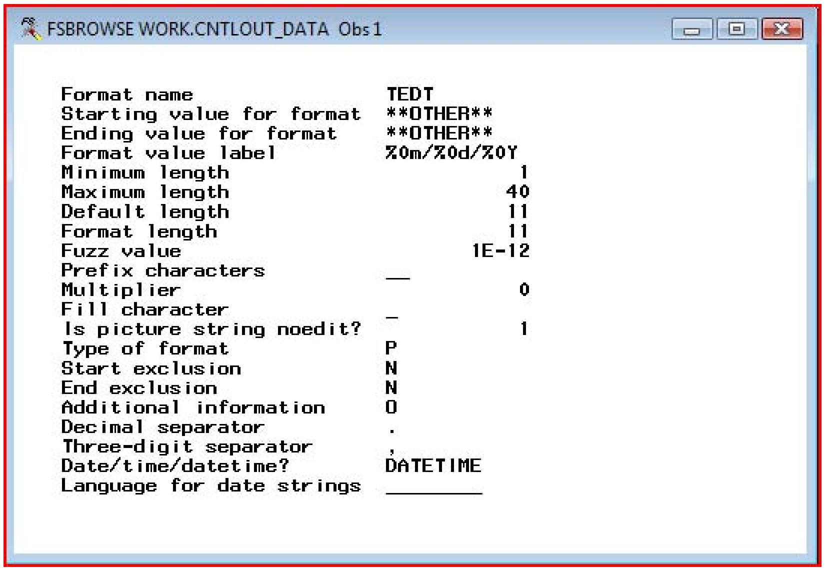
However, these commands lose any layout customization. Instead, submitting the command

```
fsbrowse cntlout_data sasuser.FSB_demo.cntlo
```

indicates that any layout customization should be kept in catalog entry `sasuser.FSB_demo.cntlo`. An important feature is supported when the form is opened by submitting code, like:

```
proc fsbrowse data= cntlout_data screen = sasuser.FSB_demo.cntlo ;
run ;
```

Perhaps you are less familiar with these columns and would like the additional info provided by the variable labels.



```
FSBROWSE WORK.CNTLOUT_DATA  Obs 1

Format name                 TEDT
Starting value for format   **OTHER**
Ending value for format     **OTHER**
Format value label          %0m/%0d/%0Y
Minimum length                        1
Maximum length                       40
Default length                       11
Format length                        11
Fuzz value                        1E-12
Prefix characters           __
Multiplier                            0
Fill character              _
Is picture string noedit?             1
Type of format              P
Start exclusion             N
End exclusion               N
Additional information      O
Decimal separator           .
Three-digit separator       ,
Date/time/datetime?         DATETIME
Language for date strings   ________
```

Figure 3. Columns described by LABEL option.

In the code submitted, only the LABEL option is added to have variable names replaced with variable labels.

```
proc fsbrowse data= cntlout_data screen = sasuser.FSB_demo.cntlo LABEL ;
run ;
```

Although tables are more often opened with a function key/command, submitting FSBROWSE procedure code is the only way that we can request that variable labels should appear on a new layout.

After we save the form layout we will be able to open it with a function key or command, like

```
FSB cntlout_data sasuser.FSB_demo.cntlo
```

To create the form layout with labels, go to menu: Tools/ Modify Screen

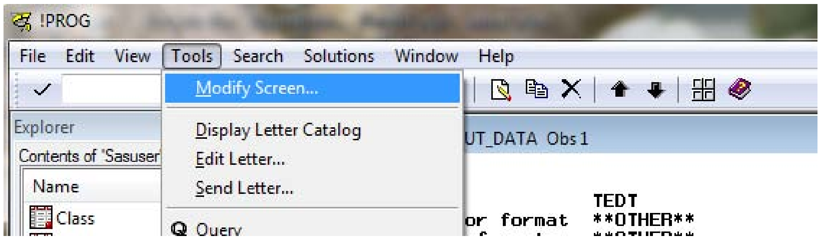
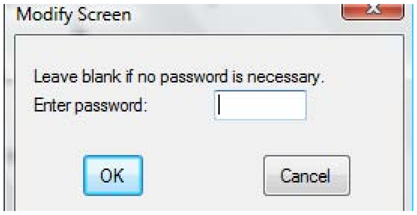


Figure 4 to create and save the form

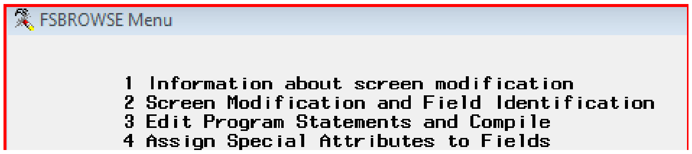
When the form is opened for the first time, FSBROWSE asks if we want a password to protect the layout. I generally ignore the prompt about passwords: just click on OK



Leave blank if no password is necessary.
Enter password:

Figure 5 password prompt

That opens another window



```
1 Information about screen modification
2 Screen Modification and Field Identification
3 Edit Program Statements and Compile
4 Assign Special Attributes to Fields
```

Figure 6. FSBROWSE Menu

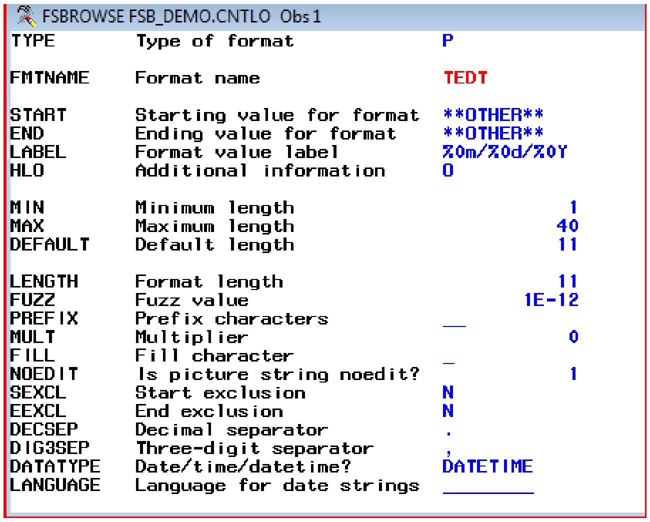
Here all we need do to save the screen layout with those labels is press the “Goback” button. The layout with the labels has been saved in the catalog.

The catalog entry is named by the code (sasuser.FSB_demo.cntlo) with object type “SCREEN”. So its formal name is SASUSER.FSB_DEMO. DEMO_CNTLO.SCREEN.

For sake of clarity FSBROWSE refers to a form layout as a “screen layout”.

ENHANCING FORM LAYOUT

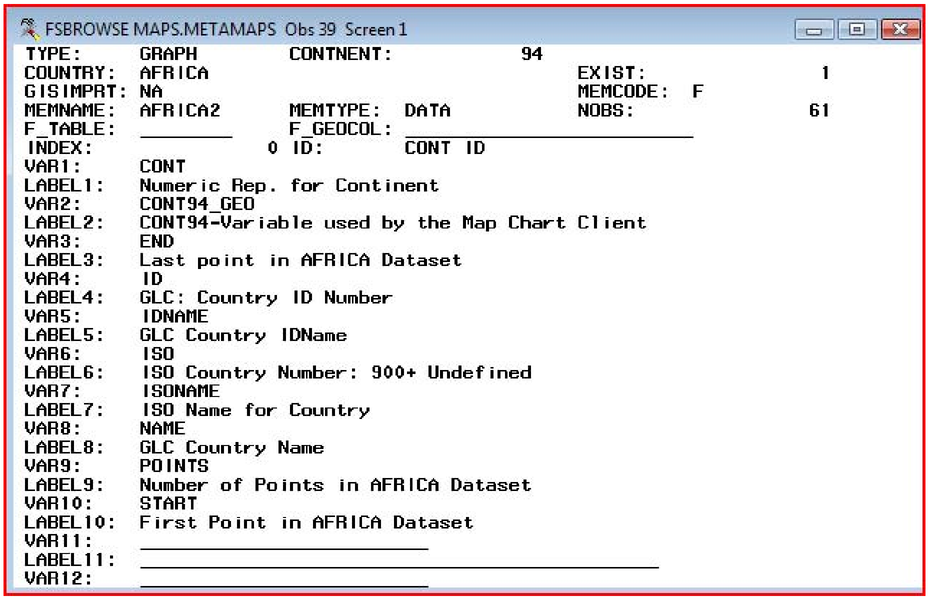
With little work, labels and variables can be combined and rearranged to look like:



```
TYPE         Type of format            P

FMTNAME      Format name               TEDT

START        Starting value for format **OTHER**
END          Ending value for format   **OTHER**
LABEL        Format value label        %0m/%0d/%0Y
HLO          Additional information    0

MIN          Minimum length                  1
MAX          Maximum length                 40
DEFAULT      Default length                 11

LENGTH       Format length                  11
FUZZ         Fuzz value                  1E-12
PREFIX       Prefix characters         __
MULT         Multiplier                      0
FILL         Fill character            _
NOEDIT       Is picture string noedit?       1
SEXCL        Start exclusion           N
EEXCL        End exclusion             N
DECSEP       Decimal separator         .
DIG3SEP      Three-digit separator     ,
DATATYPE     Date/time/datetime?       DATETIME
LANGUAGE     Language for date strings ________
```

Figure 7. Form rearranged to improve accessibility

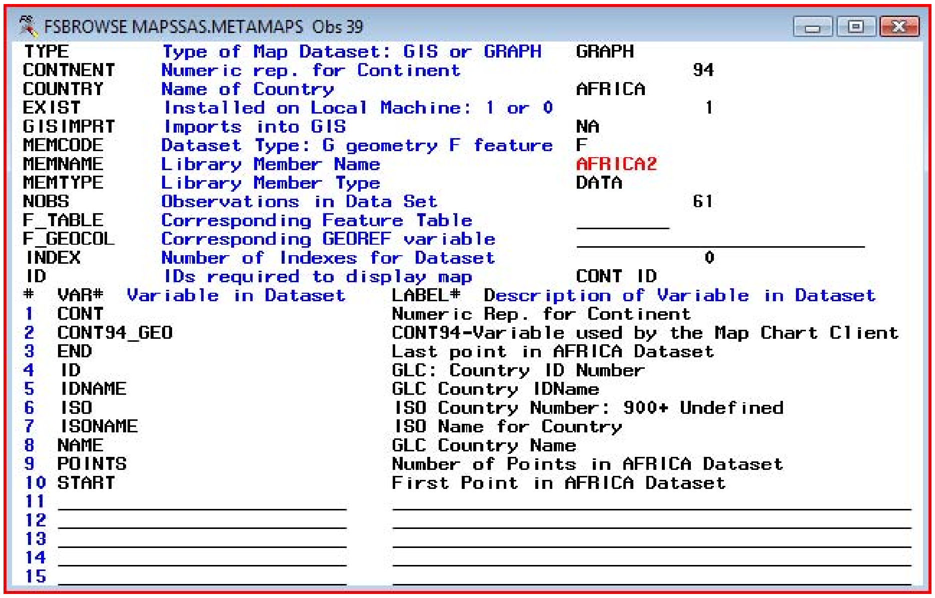
With a repeating group of fields (like de-normalized data of a data mart) here is another default layout



```
TYPE:      GRAPH        CONTNENT:            94
COUNTRY:   AFRICA                                 EXIST:               1
GISIMPRT:  NA                                     MEMCODE:  F
MEMNAME:   AFRICA2      MEMTYPE:  DATA            NOBS:               61
F_TABLE:   ________     F_GEOCOL: _________________________
 INDEX:              0  ID:       CONT ID
VAR1:      CONT
LABEL1:    Numeric Rep. for Continent
VAR2:      CONT94_GEO
LABEL2:    CONT94-Variable used by the Map Chart Client
VAR3:      END
LABEL3:    Last point in AFRICA Dataset
VAR4:       ID
LABEL4:    GLC: Country ID Number
VAR5:       IDNAME
LABEL5:    GLC Country IDName
VAR6:       ISO
LABEL6:     ISO Country Number: 900+ Undefined
VAR7:       ISONAME
LABEL7:     ISO Name for Country
VAR8:      NAME
LABEL8:    GLC Country Name
VAR9:      POINTS
LABEL9:    Number of Points in AFRICA Dataset
VAR10:     START
LABEL10:   First Point in AFRICA Dataset
VAR11:     _________________________
LABEL11:   _____________________________________________
VAR12:     _________________________
```

Figure 8. default layout for a table with repeating group of columns

which can be expanded with variable labels and rearranged as a table like :

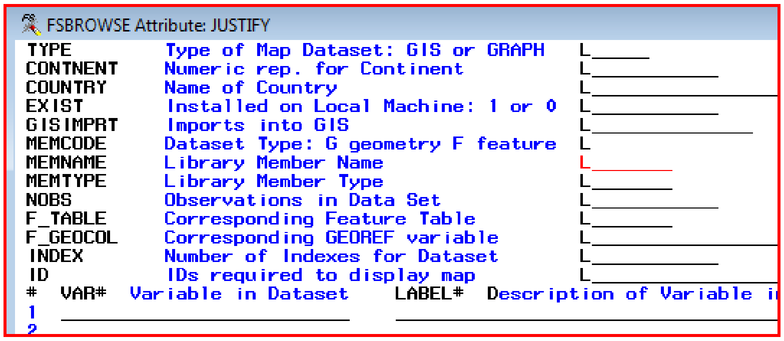


```
TYPE         Type of Map Dataset: GIS or GRAPH   GRAPH
CONTNENT     Numeric rep. for Continent                    94
COUNTRY      Name of Country                     AFRICA
EXIST        Installed on Local Machine: 1 or 0             1
GISIMPRT     Imports into GIS                    NA
MEMCODE      Dataset Type: G geometry F feature  F
MEMNAME      Library Member Name                 AFRICA2
MEMTYPE      Library Member Type                 DATA
NOBS         Observations in Data Set                      61
F_TABLE      Corresponding Feature Table         ________
F_GEOCOL     Corresponding GEOREF variable       _________________________
INDEX        Number of Indexes for Dataset                  0
ID           IDs required to display map         CONT ID
#  VAR#  Variable in Dataset        LABEL#  Description of Variable in Dataset
1  CONT                             Numeric Rep. for Continent
2  CONT94_GEO                       CONT94-Variable used by the Map Chart Client
3  END                              Last point in AFRICA Dataset
4  ID                               GLC: Country ID Number
5  IDNAME                           GLC Country IDName
6  ISO                              ISO Country Number: 900+ Undefined
7  ISONAME                          ISO Name for Country
8  NAME                             GLC Country Name
9  POINTS                           Number of Points in AFRICA Dataset
10 START                            First Point in AFRICA Dataset
11 _________________________        _____________________________________________
12 _________________________        _____________________________________________
13 _________________________        _____________________________________________
14 _________________________        _____________________________________________
15 _________________________        _____________________________________________
```

Figure 9. table with groups into columns

Here, the purpose is not to reveal the metadata of SAS/GIS® software nor SAS/GRAPH® maps, but to show how improving on default arrangement pays dividends – and simply when improvement is incremental.

(as I appreciate consistency of layout as well as simplicity, here is a simple enhancement)

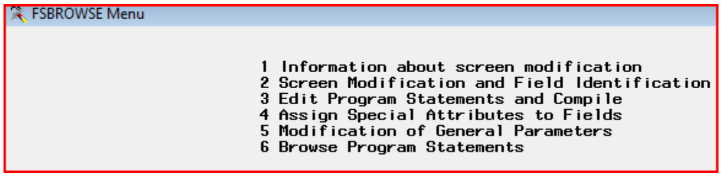


```
TYPE         Type of Map Dataset: GIS or GRAPH   L_____
CONTNENT     Numeric rep. for Continent          L___________
COUNTRY      Name of Country                     L_________________
EXIST        Installed on Local Machine: 1 or 0  L___________
GISIMPRT     Imports into GIS                    L______________
MEMCODE      Dataset Type: G geometry F feature  L
MEMNAME      Library Member Name                 L_______
MEMTYPE      Library Member Type                 L_______
NOBS         Observations in Data Set            L___________
F_TABLE      Corresponding Feature Table         L_______
F_GEOCOL     Corresponding GEOREF variable       L_________________
INDEX        Number of Indexes for Dataset       L___________
ID           IDs required to display map         L_________________
#  VAR#  Variable in Dataset        LABEL#  Description of Variable i
1  _________________________        ___________________________________
2
```

Figure 10. setting Justify attribute

Here I am defining that I would like all these values to be (L) left justified.

Justification is among the settings controlled by option 4 “Assign Special Attributes” on the FSBROWSE Menu panel

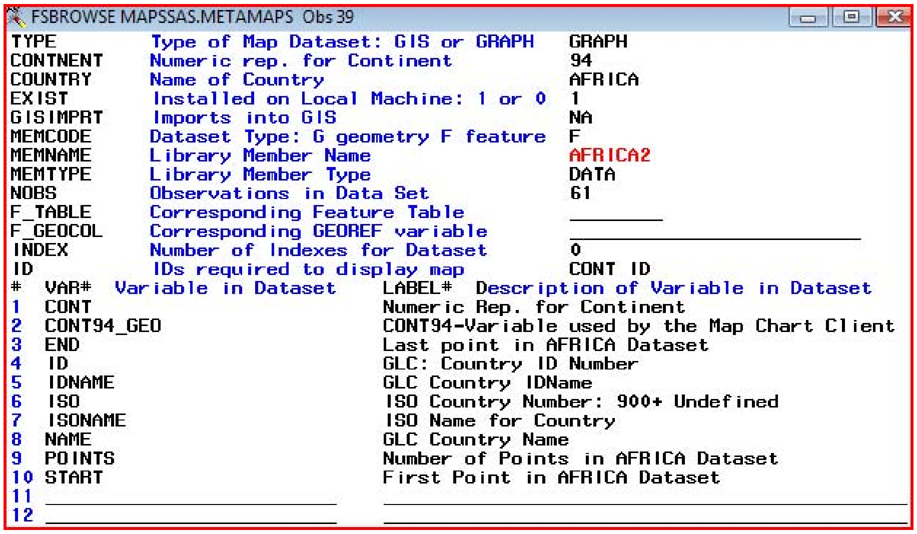


```
1 Information about screen modification
2 Screen Modification and Field Identification
3 Edit Program Statements and Compile
4 Assign Special Attributes to Fields
5 Modification of General Parameters
6 Browse Program Statements
```

Figure 11. menu to start changing Special Attributes

I wanted to use this to demonstrate incremental improvement. This is the effect

274-2012: SAS/FSP(r) Provides the Best Customizable Form Viewer, continued



FSBROWSE MAPSSAS.METAMAPS Obs 39

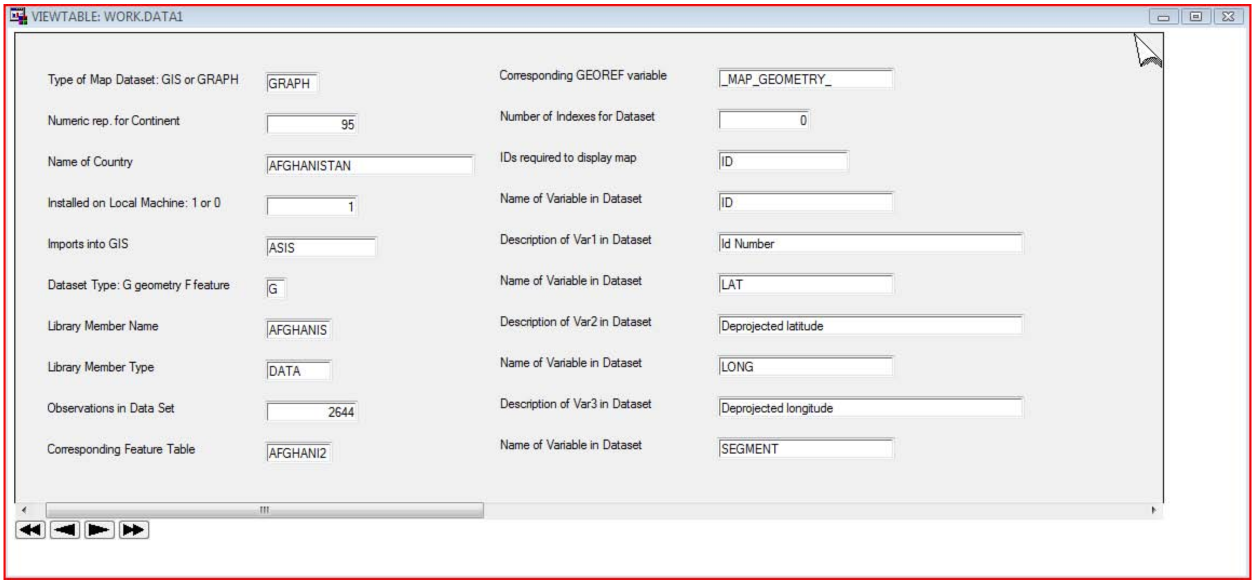
```
TYPE        Type of Map Dataset: GIS or GRAPH    GRAPH
CONTNENT    Numeric rep. for Continent           94
COUNTRY     Name of Country                      AFRICA
EXIST       Installed on Local Machine: 1 or 0   1
GISIMPRT    Imports into GIS                     NA
MEMCODE     Dataset Type: G geometry F feature   F
MEMNAME     Library Member Name                  AFRICA2
MEMTYPE     Library Member Type                  DATA
NOBS        Observations in Data Set             61
F_TABLE     Corresponding Feature Table          ________
F_GEOCOL    Corresponding GEOREF variable        _________________________
INDEX       Number of Indexes for Dataset        0
ID          IDs required to display map          CONT ID
#  VAR#  Variable in Dataset    LABEL#  Description of Variable in Dataset
1  CONT                         Numeric Rep. for Continent
2  CONT94_GEO                   CONT94-Variable used by the Map Chart Client
3  END                          Last point in AFRICA Dataset
4  ID                           GLC: Country ID Number
5  IDNAME                       GLC Country IDName
6  ISO                          ISO Country Number: 900+ Undefined
7  ISONAME                      ISO Name for Country
8  NAME                         GLC Country Name
9  POINTS                       Number of Points in AFRICA Dataset
10 START                        First Point in AFRICA Dataset
11 _________________________    ____________________________________________
12 _________________________    ____________________________________________
```

Figure 12. Layout updated - always start values in the same column

ALTERNATIVE FORM VIEWERS

LOOKING INSIDE THE SAS WORLD :

1. Display Manager Viewtable form view



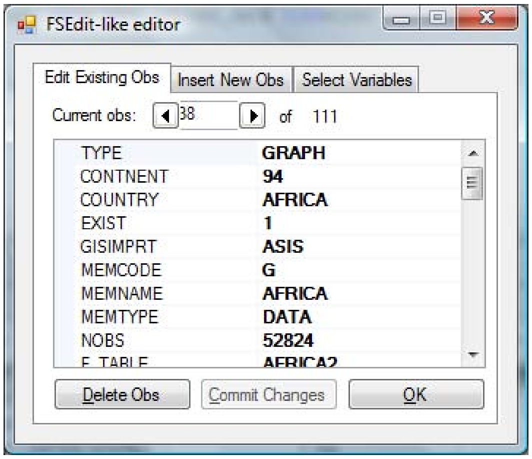
VIEWTABLE: WORK.DATA1

Type of Map Dataset: GIS or GRAPH	GRAPH	Corresponding GEOREF variable	_MAP_GEOMETRY_
Numeric rep. for Continent	95	Number of Indexes for Dataset	0
Name of Country	AFGHANISTAN	IDs required to display map	ID
Installed on Local Machine: 1 or 0	1	Name of Variable in Dataset	ID
Imports into GIS	ASIS	Description of Var1 in Dataset	Id Number
Dataset Type: G geometry F feature	G	Name of Variable in Dataset	LAT
Library Member Name	AFGHANIS	Description of Var2 in Dataset	Deprojected latitude
Library Member Type	DATA	Name of Variable in Dataset	LONG
Observations in Data Set	2644	Description of Var3 in Dataset	Deprojected longitude
Corresponding Feature Table	AFGHANI2	Name of Variable in Dataset	SEGMENT

Figure 13. ViewTable form view

2. Enterprise Guide “observation editor addin”

The default presentation looks like

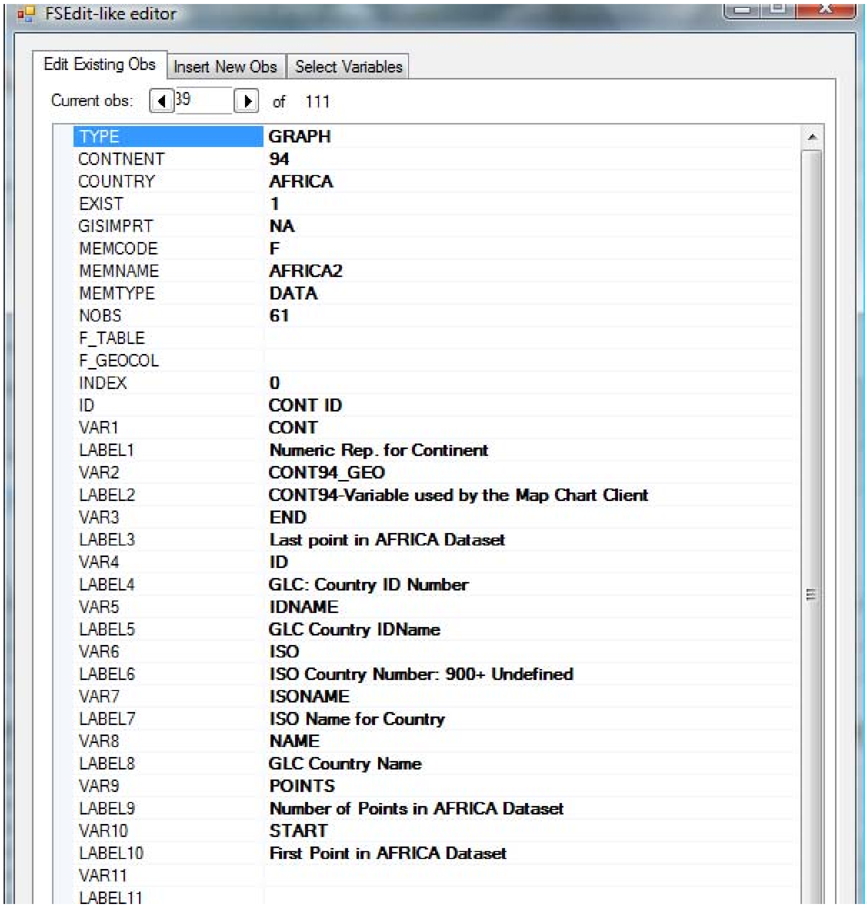


Current obs: 38 of 111

TYPE	GRAPH
CONTNENT	94
COUNTRY	AFRICA
EXIST	1
GISIMPRT	ASIS
MEMCODE	G
MEMNAME	AFRICA
MEMTYPE	DATA
NOBS	52824
F_TABLE	AFRICA2

Figure 14. Enterprise Guide "observation editor addin"

which with a little point-n-click can be improved to



Current obs: 39 of 111

TYPE	GRAPH
CONTNENT	94
COUNTRY	AFRICA
EXIST	1
GISIMPRT	NA
MEMCODE	F
MEMNAME	AFRICA2
MEMTYPE	DATA
NOBS	61
F_TABLE	
F_GEOCOL	
INDEX	0
ID	CONT ID
VAR1	CONT
LABEL1	Numeric Rep. for Continent
VAR2	CONT94_GEO
LABEL2	CONT94-Variable used by the Map Chart Client
VAR3	END
LABEL3	Last point in AFRICA Dataset
VAR4	ID
LABEL4	GLC: Country ID Number
VAR5	IDNAME
LABEL5	GLC Country IDName
VAR6	ISO
LABEL6	ISO Country Number: 900+ Undefined
VAR7	ISONAME
LABEL7	ISO Name for Country
VAR8	NAME
LABEL8	GLC Country Name
VAR9	POINTS
LABEL9	Number of Points in AFRICA Dataset
VAR10	START
LABEL10	First Point in AFRICA Dataset
VAR11	
LABEL11	

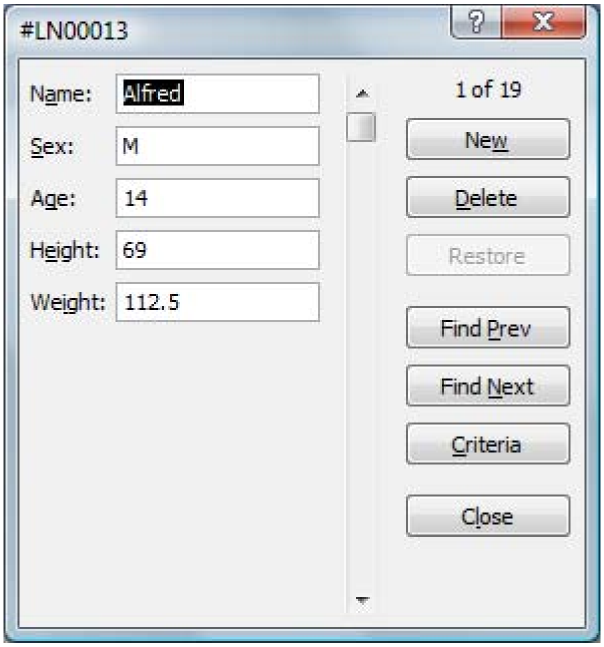
Figure 15. Enterprise Guide "observation editor addin" extended

However, there seems no way to set this as the default - so each time we spend a more time adjusting the layout – or give up in the attempt!

LOOKING OUSIDE THE SAS WORLD :

Microsoft Excel offers a “Form” option on the “Data” menu of Excel 2003 (hidden from the Data ribbon in Excel 2007 (and 10?) – but it is still available), but with some limitations.

Here selecting “View in Excel” for SASHELP.CLASS (and clicking an icon for Excel’s Form View) demonstrates..



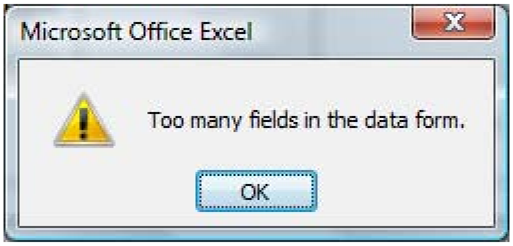
#LN00013

Name: Alfred
Sex: M
Age: 14
Height: 69
Weight: 112.5

1 of 19

Figure 16. SASHELP.CLASS in Excel Form View

However, for SASHELP.VTABLE, selecting the “Form” icon, I get



Microsoft Office Excel

Too many fields in the data form.

Figure 17. SASHELP.VTABLE in Excel Form View

The limit is 32 columns in Excel releases up to Office 2007. I need to layout many more columns than this limit. Of course I could layout an Excel worksheet as a Form, and pull in data from the Excel “get external data query”, but it adds up to more work, and I’ve not found a practical and generic way of integrating this with SAS (yet?).

INTEGRATING FSBROWSE FORM VIEWER WITH SAS

As I hope I said earlier, FSBROWSE depends on the SAS Display Manager environment. It can be used in the default way to view data sets in the SAS explorer, through the code and commands demonstrated earlier. This paper shows how the SAS Explorer can be extended to use FSBROWSE, almost seamlessly – as follows:

Navigate through the SAS Explorer to directory window of a SAS library. The following slides show my WORK library with a table generated with the code

```
proc sql ;
   create table vtable as select * from dictionary.tables
          where libname in( 'SASHELP' 'MAPS' ) and nvar > 30 ;
quit;
```

I have "right-clicked" on the table "Vtable"

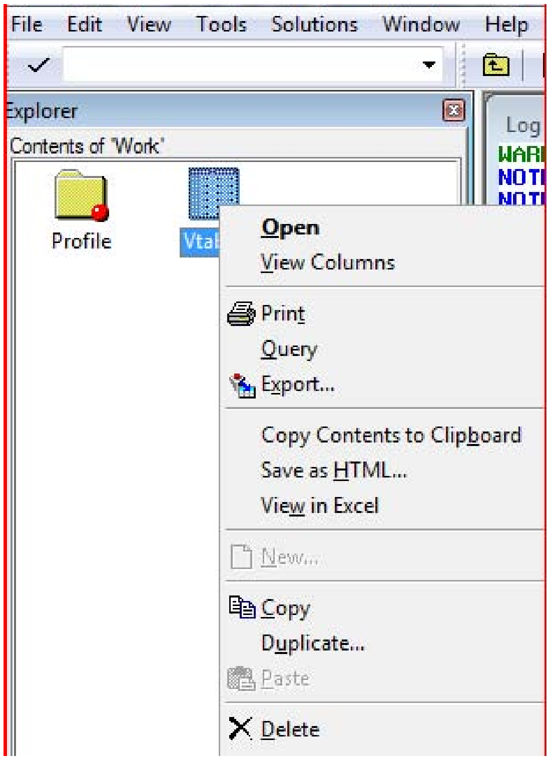
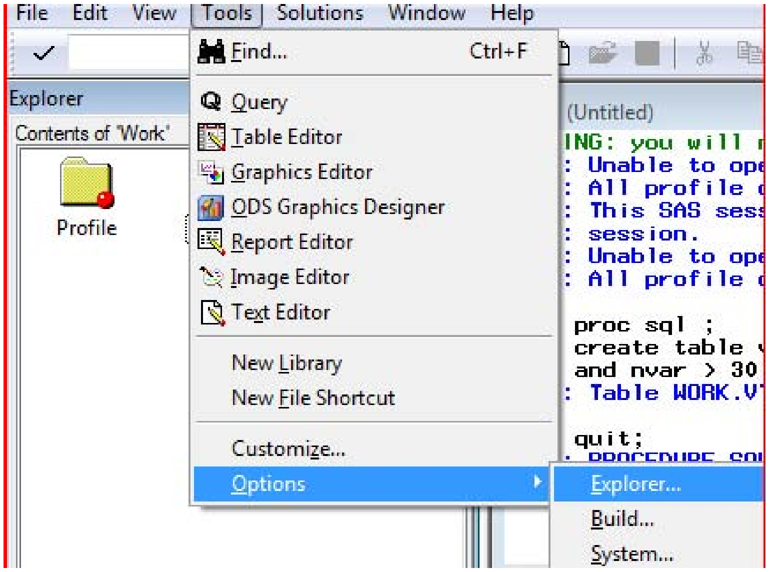


Figure 18. default menu for a table in SAS Explorer

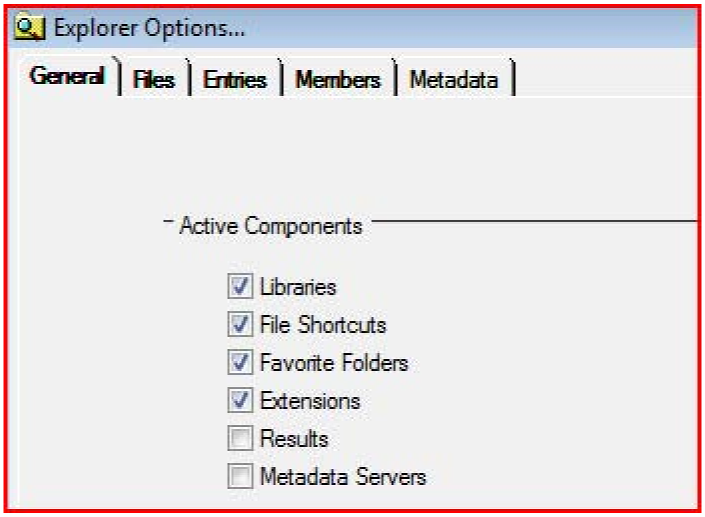
Although FSBROWSE does not appear on this pop-up menu, the menu can be extended with code or "point-and-click" – as follows. Starting in a SAS explorer window (like this window into my WORK library), drill down the "Tools" menu as follows



```
ING: you will r
: Unable to ope
: All profile c
: This SAS sess
: session.
: Unable to ope
: All profile c

 proc sql ;
 create table w
 and nvar > 30
: Table WORK.V

 quit;
```

Figure 19. extending SAS Explorer – 1

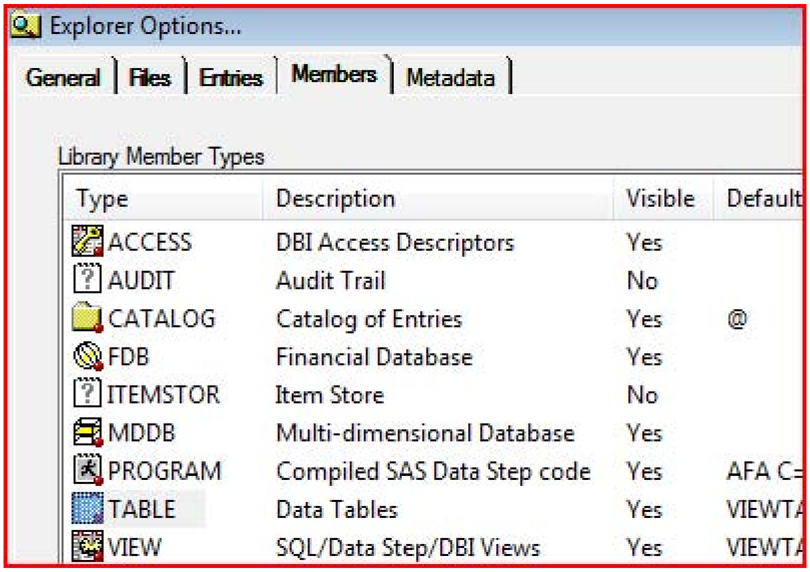
Selecting “Options” takes us into this dialog



☑ Libraries
☑ File Shortcuts
☑ Favorite Folders
☑ Extensions
□ Results
□ Metadata Servers

Figure 20. extending SAS Explorer – 2

Select the “Members” tab (we want to add provide a Form View of a SAS data set which is a “library member” type of metadata)



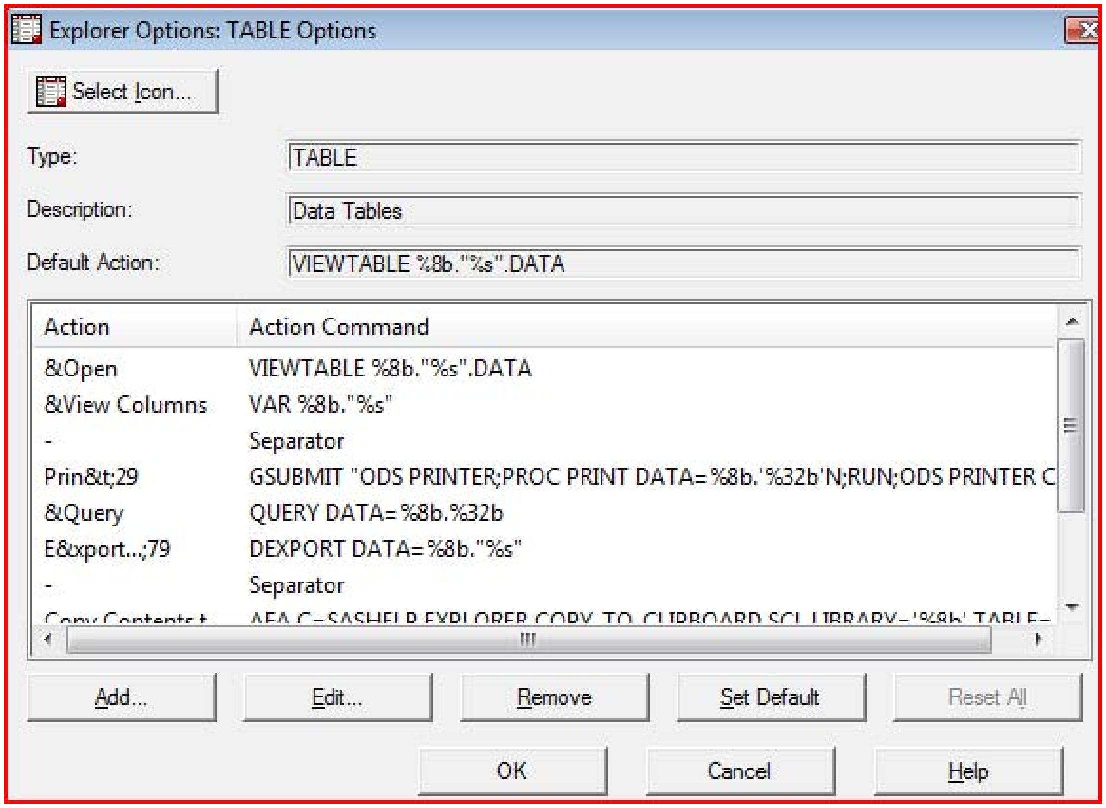
Library Member Types

Type	Description	Visible	Default
ACCESS	DBI Access Descriptors	Yes	
AUDIT	Audit Trail	No	
CATALOG	Catalog of Entries	Yes	@
FDB	Financial Database	Yes	
ITEMSTOR	Item Store	No	
MDDB	Multi-dimensional Database	Yes	
PROGRAM	Compiled SAS Data Step code	Yes	AFA C=
TABLE	Data Tables	Yes	VIEWTA
VIEW	SQL/Data Step/DBI Views	Yes	VIEWTA

Figure 21. extending SAS Explorer – 3

and double-click on the "TABLE" row (in the Type column).

This lists the commands that will be issued for each of the menu items offered when you right-click an item of this "Type" in the SAS Explorer.



Type: TABLE

Description: Data Tables

Default Action: VIEWTABLE %8b."%s".DATA

Action	Action Command
&Open	VIEWTABLE %8b."%s".DATA
&View Columns	VAR %8b."%s"
-	Separator
Prin&t;29	GSUBMIT "ODS PRINTER;PROC PRINT DATA=%8b.'%32b'N;RUN;ODS PRINTER C
&Query	QUERY DATA=%8b.%32b
E&xport...;79	DEXPORT DATA=%8b."%s"
-	Separator
Copy Contents t	AFA C=SASHELP.EXPLORER.COPY_TO_CLIPBOARD.SCL LIBRARY='%8b' TABLE=

Figure 22. extending SAS Explorer – 4

We want to add another action - the invocation of FSBROWSE, so we press the "Add..." button.

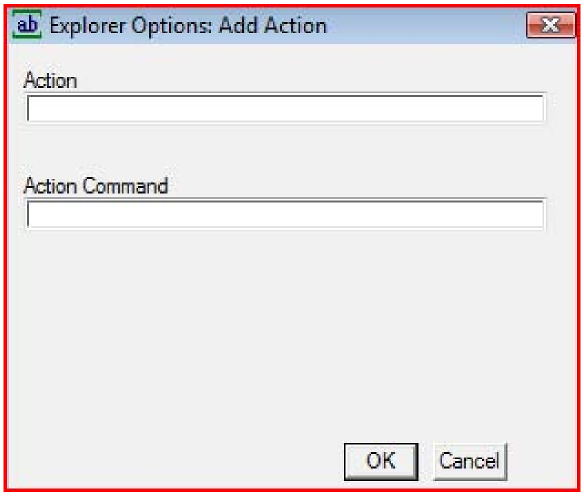


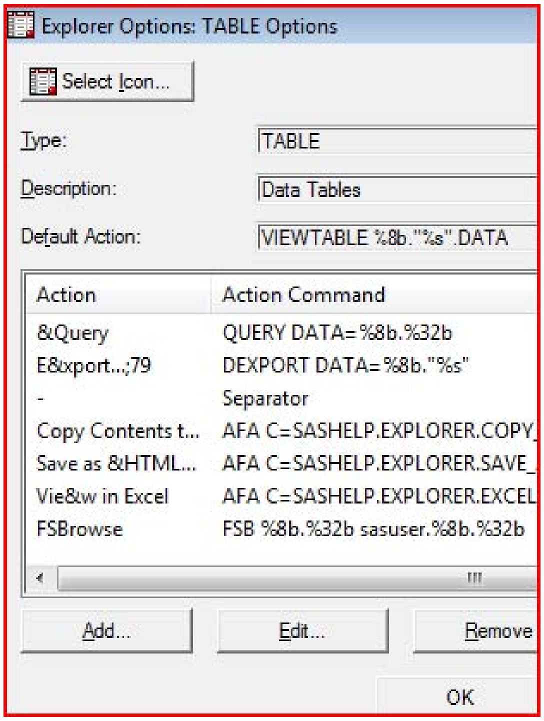
Figure 23. extending SAS Explorer – 5

In the “Explorer Options: Add Action” form, fill the “Action“ field with “FSBrowse”, and the “Action Command” field with

```
   FSB %8b.%32b sasuser.%8b.%32b
```

and click OK (or just press “enter” key).

That returns us to the earlier dialog where by scrolling down in the list box, we can see the new action that we have just defined, now appears last on the list.



Type: TABLE

Description: Data Tables

Default Action: VIEWTABLE %8b."%s".DATA

Action	Action Command
&Query	QUERY DATA=%8b.%32b
E&xport...;79	DEXPORT DATA=%8b."%s"
-	Separator
Copy Contents t...	AFA C=SASHELP.EXPLORER.COPY_
Save as &HTML...	AFA C=SASHELP.EXPLORER.SAVE_
Vie&w in Excel	AFA C=SASHELP.EXPLORER.EXCEL
FSBrowse	FSB %8b.%32b sasuser.%8b.%32b

Figure 24. extending SAS Explorer – 6

We need to accept these changes by clicking OK twice more.

Now the behavior has been stored in the SAS Registry.

We can make a “back up” of “user customization” of the SAS Registry, with code like

```
    proc REGISTRY export= "%sysget(mysasfiles)\my_reg.&sysdate9..txt" ;
    run ;
    proc FSLIST     file= "%sysget(mysasfiles)\my_reg.&sysdate9..txt" ;
    run ;
```

Figure 25. extending SAS Explorer – 7

The content of the text file created by the REGISTRY Procedure (viewed here, with the FSLIST procedure), provides interesting insight to the SAS registry resource, and points out how another option of PROC REGISTRY, IMPORT= might be used to bypass the point-and-click interface and just import our updates for the SAS Explorer.

Now demonstrate integration FSBROWSE in the SAS Explorer. Right-click on a table in SAS Explorer, and:

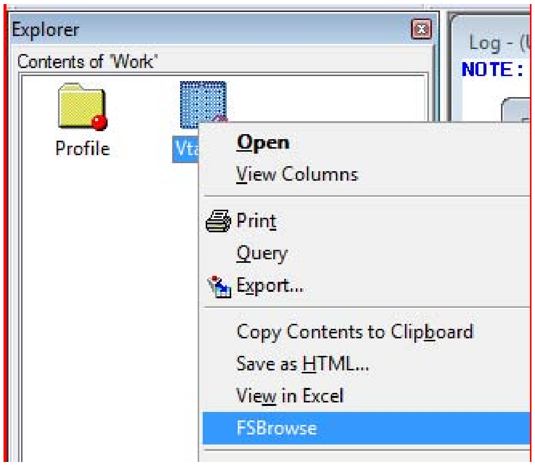


Figure 26. extending SAS Explorer – 8 – demonstrate pop-up menu is extended

Selecting “FSBrowse” from the menu invokes the command

```
FSBrowse WORK.VTABLE SASUSER.WORK.VTABLE
```

(because the table showing in the work library is named Vtable)

As stated in the documentation for the FSBROWSE command, this command will open the SAS data set with the optional parameter which names a screen data set in which new screen layout customization will be saved – and saved into the catalog SASUSER.{libname} with the object (or entry) name {memname} i.e. SASUSER.WORK.VTABLE. Of course, if the entry already exists, any customization it contains will be used to control how an observation of data will be displayed.

Any time in future when I have a table of that name (vtable) in that library (work), this option on our updated pop-up menu will use our customized screen layout

OTHER FSBROWSE VIEW CAPABILITIES

FSBROWSE provides one facility that seems unavailable elsewhere among SAS data presentation facilities. Direct addressing of data rows is supported. When you ask the viewer to show the last row, it reads that row into the form view directly. The same action in other viewers results in the viewer program reading all rows until the requested row appears. To immediately navigate to the millionth row of a wide table and with sub-second response time is un-matched.

FSBROWSE provides more than one way of searching for rows within the table. As well as “WHERE”, “FIND”, “LOCATE” and “SEARCH” commands are supported. The WHERE operates as in Viewtable and will use any indexes on the table. Although LOCATE, SEARCH and FIND do not use indexes, their behavior is subtly different and useful. The SEARCH@ command facilitates “full contents searching” across more than one column.

Additional columns can be added to a screen

With the (original version of) SAS/SCL, additional columns added to a screen, can be derived, for example:

- to show both formatted and unformatted values of a variable;

- to calculate and show an “age today” next to a “date of birth”;
- to repeat key fields on “over-flow” screens.

Layout customization is limited to strings and does not include graphics (not always a disadvantage).

CONCLUSION

When investigating details about an observation, many columns may need organizing on screen. FSBROWSE offers extensive ways to layout data on a screen. For those environments which cannot use FSBROWSE (because these environments like Enterprise Guide, do not use SAS Display Manager) the concepts may prove useful to establish the requirements for an alternative viewer. Within the Display Manager environment, the SAS Explorer can be extended to offer “FSBrowse” action for a SAS data set.

REFERENCES

- SAS Institute Inc. 2011. SAS/FSP® 9.3 Procedures Guide. Cary, NC: SAS Institute Inc.
- DeVenezia,Richard. “SAS Explorer Actions (last updated 17 February 2005 but still relevant). Available at http://www.devenezia.com/downloads/sas/actions/.

ACKNOWLEDGMENTS

Richard DeVenezia hosts a website with useful information about SAS Explorer Actions

RECOMMENDED READING

- SAS/FSP® 9.3 Procedures Guide
- Richard DeVenezia’s website at the section on SAS Explorer Actions at http://www.devenezia.com/downloads/sas/actions/

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author on the SAS Community for this paper http://www.sascommunity.org/wiki/FSB_the_best_Form_Viewer or at:

Name: Peter Crawford
Enterprise: Crawford Software Consultancy Limited
Address: London, UK
Phone: +44(0) 7802 732254
E-mail: Peter.Crawford@blueyonder.co.uk