

Paper 027-29

## The Q & A's of SAS/AF<sup>®</sup> Development

Bernd E. Imken, Patented Medicine Prices Review Board, Ottawa, ON, Canada

### ABSTRACT

This paper focuses on the use of SAS/AF software in the development of strategic applications. It will show how you can build user friendly applications utilizing push buttons, text boxes, pull down lists and graphical displays etc.; how you can customize the appearance of an application to enhance appearance, data integrity, or usability; and how you can utilize drag and drop within an application.

The "Frequently Asked Questions" approach focuses on typical difficulties encountered during applications development. SAS/AF developers will discover how to build applications using Form and Table Viewers to create a graphical user interfaces to view or edit SAS data. Particular emphasis will be placed on the customization of such viewers and on how viewers can automatically be linked together without coding. Examples of developing Master/Detail Applications will also be thoroughly discussed. A similar paper was presented at Sugi27 entitled "Developing Master/Detail Applications Using Form & Table Viewers." That paper won the Best Contributed Paper award in the Sugi27 Applications Development Section. This paper extends the original, adding many additional Questions and Answers regarding SAS/AF development.

### INTRODUCTION

This paper spotlights problems that I have encountered in developing applications using SAS<sup>®</sup> Release 8.2. This release had a significant number of improvements and fixes applied that were not available with Release 8.1. SAS<sup>®</sup> has traditionally offered numerous ways of viewing full screen data. For many years, application developers used procedures such as FSEDIT, FSBROWSE, FSVIEW and FSLIST. Most recently, SAS<sup>®</sup> added VIEWTABLE. Although easy to use, these procedures lacked the high level of customization demanded by developers. SAS/AF now enables developers to create graphical user interfaces similar to the best available within a Windows environment. The recent addition of FORM VIEWER and TABLE VIEWER Visual Components now allows greater use of graphics, drag & drop, multiple fonts etc. In addition, you can now link these new components to provide Master/Detail views within the same viewing frame.

### QUESTION #1 What is SAS/AF? How does it differ from other SAS products such as the SAS Application Development Studio and software such as Web/AF?

**ANSWER #1** SAS/AF is used to develop applications which would be used in a native SAS environment either on a client computer or on a LAN/WAN. Other SAS tools such as the Application Development Studio are design for Internet/Intranet usage.

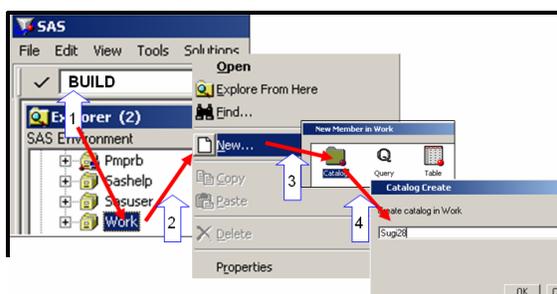
### Q2 What SAS software is required?

**A2** In order to develop or modify applications SAS/AF must be licensed on your computer. The Base SAS license is of course also required. Other licenses such as SAS/Graph may be required if the developer uses them within the application. Users can run previously developed SAS/AF applications on their computers without a SAS/AF license on their machines. They do however need a license for Base SAS as well as any other packages such as SAS/Graph which may have been used in developing the application.

### Q3 HOW DO I DEVELOP MY FIRST SAS/AF FRAME?

**A3.A** The SAS/AF Frame is the equivalent to a page or screen within an application. Usually an AF application is made up of a series such Frame's which are logically connected to each other. For example a very simple application may be comprised of a: Title page, a report menu selection page and closing screen. Through the rest of this paper I will use the terminology Frame rather than page or screen.

First, you must develop a new catalogue in which to place your application (or select an existing catalogue if one had previously been established. The diagram to the left shows one method of developing such a catalogue.

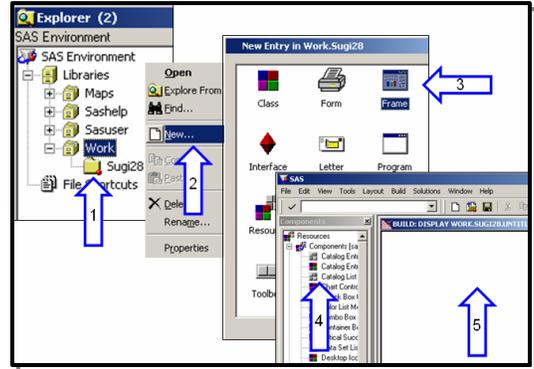


1. Type BUILD on the command line to open the Explorer window.
2. Right click on an existing Library to open the second window show on the left.

3. Select "New" to open up the "new Member" screen.
4. Click on the Catalogue to open the Catalogue Create screen
5. Type in the name of the new Catalogue in the spot provided.

**A3.b** Next, you can go about creating the Frame within the Catalogue.

1. Select the catalogue in which you want the place the new Frame. Right Click the catalogue icon.
2. Select "New" from the pop up screen.
3. Select Frame from the screen entitled "New Entry to .."
4. the new Frame has just been created. Notice in the diagram on the right that the Frame initially is blank and is located on the right hand side of the screen. On the left side of the screen you will see the Component window displayed.



**Alternate Approach:** It is also possible to create a catalog directly and a new Frame at the same time by typing in the following command within the SAS Command line/box.

```
BUILD work.sugi29.demo.frame
```

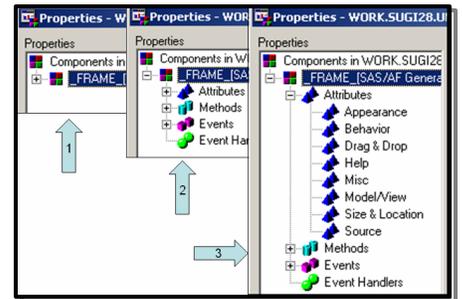
Or, you could have submitted the following SAS code from the Program Editor.

```
Proc build c=work.sugi29.demo.frame;run;
```

**Q4 Can I change the appearance of this new FRAME?**

- A4
  - Right Click anywhere on the blank Frame
  - Click on "Properties"
  - Click on `_FRAME_`
  - Click on + sign to the left of `_FRAME_`
  - Click on Attributes
  - Click on Appearance

**FRAME?**

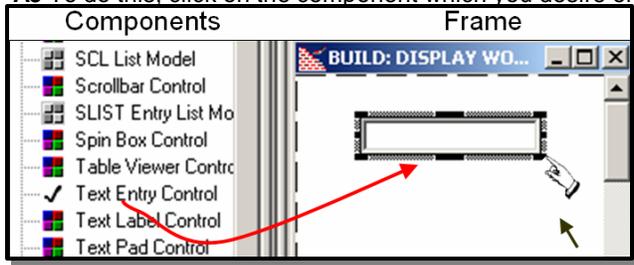


**Frequently Accessed `_Frame_` Component Attributes**

Type	Attribute	Selection	Explanation
Appearance	<code>backgroundColor</code>	Black, Blue, Brown, Cyan, Gray etc.	Most colors can be selected.
	<code>bannerType</code>	Command, Select, None	Most commonly "None" is selected
	<code>title</code>	Any title you desire	Your choice
Behavior	<code>icon</code>	Select icon number	Changes the icon  on the top left of the Frame
	<code>keysEntry</code>	Specify location of keys entry in the SASUSER PROFILE catalog	Modify your Function keys by typing "keys" in the command box while in Build mode. Then select "Save AS" from under the File menu
	<code>pmenuEntry (very useful)</code>	Specify location of pmenu in the catalog	New Pmenus can be developed using Proc Pmenu see SAS Help for an explanation
	<code>type</code>	Standard or Dialog	Most frames are Standard
Source	<code>SCLEntry</code>	Specify location of the Frame's SCL code	SCL will be discussed in the next section.

**Q5 How can I add objects/components to this new FRAME?**

**A5** To do this, click on the component which you desire on the left and drag it over the frame and drop it in the desired location. Or, you can double click the component and the component will be opened at the top left corner of the Frame.



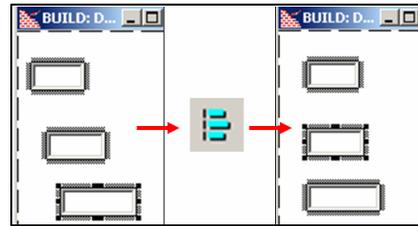
The component can be sized by dragging the sizing points on the corners of the component or midway on the horizontal or vertical lines surrounding the component.

**Q6** How can I change the appearance of a component once it has been created?

**A6** Right click on the component and the menu shown in the figure to the left will be displayed. Follow the same approach as was used to change the appearance of the frame in a previous question.

**Q7** How can I align components?

**A7** Easy, activate multiple components as shown in the figure to the right and the press the alignment icon as shown in the figure. Note there is also an alignment icon to align components horizontally.



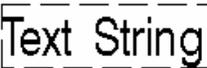
**Q8** What is SCL and how do I use it?

**A8** SCL, or SAS component Language is code which is written to control the actions on the frame. For example if the user presses a Push Button component on the screen you may wish the frame to execute a few lines of SAS code, or you may wish the user to be moved to another screen. Things which you can do by writing SCL code is almost endless. SCL code changed significantly from SAS Version 6 to SAS Version 8. In Version 6 and before it used to be called Screen Control Language. The fundamental concepts behind the older SCL has been retained within the new Version 8 edition, however numerous new features have been added such as dot notation. These newer concepts make SCL coding far simpler. This paper will not dig deeply in SCL coding due to length restrictions. There are however numerous other papers and texts available from SAS on this subject.

**Q9** What kind of components are available in SAS that I could use within my application?

**A9** SAS has many components available for you usage including:

Most Frequently Used Components:

Component Name	Component Appearance	Comments or Main Attribute (MA)
Check Box	<input checked="" type="checkbox"/> Check Box	MA: <code>selected = "Yes"</code>
Combo Box	<input type="text"/>	MA: <code>selectedItem = ''</code> , supports text completion
Desktop Icon		Requires double click, more text than push button
Graphic Text		Similar to Label control generally used for large titles
List Box		MA: <code>selectedItem = ''</code> , also supports multiple selections, uses more space than combo box
Push Button		Can add Icons and orient text
Radio Button		MA: <code>selectedItem = ''</code> , select one item from many possible choices
Text Entry	<input type="text"/>	MA: <code>text = ' '</code> , may specify char, numeric etc, only 1 line
Text Label	Label	Generally used to label text entry controls etc.



SAS also still supports most of the components which were originally developed with Version 6. here are two which I still use which are not directly available in Version 8.

#### Version 6 Objects

Object Name	Object Appearance	Comments
Tab Layout		Extremely powerful Object, allows you to place many controls on the frame, all of which are addressable by the Frame SCL.
Video Player		Used to play videos

**Q10: I would like to fill the values within a list box. I know how to add to the list box by selecting the “items” attribute, but how do I get the list box to obtain the items from a SAS® data set instead?**

**A10:** If you want to populate a list box for a data set, simply click on the “variable Values List Model” and drag it over the list box just like you overlaid the model when creating a Form Viewer. Next click on the variable values list model “variablevalueslist1” and click on dataSet attribute to associate it with a SAS® data set. After that, set the variable attribute. It’s easy!

**Q11: How can I create a computed column within a Form Viewer or Table Viewer?**

**A11:** Click on the “columns” attribute of the data set model, you will see a “computed” button on the bottom left of the screen. After pressing it you will be shown another screen allowing you to add new computed columns.

**Q12: I find it very inefficient to have to click on each object and set attributes. For example, is there a way I could set all my text entries to be a specific width?**

**A12:** Windows and SAS® give you an easy way to do it. Whenever you click on the CTRL button and hold it down and then select multiple controls – all remain selected. When you now right click on any selected



item and for example change the width property, all those highlighted will take that value. Or, after selecting multiples you

could have clicked in the Properties icon on the toolbar at the top of the screen. The first 4 icons are used to align objects, the next takes you to the Component window and the last takes you to Properties.

**Q13: I deleted the text box originally created by the Form Viewer and created a combo box in its place. When I test my application the value in the box is not displayed. Why?**

**A13:** Whenever you delete an object from the Viewer, you also at the same time delete the linkage between the Viewer and the SAS® data set. If you want to link a new Viewer object, such as combo box, you must re-establish the link with the data set by right clicking on the Viewer, selecting “Form ->”, then selecting “Display column window”. On this window click on the variable you want to re-link and drag it over the new combo box.

This last item is discussed in the following SAS® Note.

<http://www.sas.com/service/techsup/unotes/SN/004/004928.html>

**Q14: How can I determine if a screen object is linked to a data set variable?**

**A14:** Right click on the Form Viewer and select “Form ->” then select “Turn Column Describe On”. Next, click on the variable you want to check. A message will be sent to the LOG telling you if the Viewer object is linked to the column (or variable) in the data set.

**Q15: I have created a push button beside a Form Viewer text entry called “Location”. When I click the push button, I want the text entry to be set to “Orlando”. How do I do it?**

**A15:** After you have created the pushbutton, you must write the following SCL code and store it in the SCL of the Form Viewer:

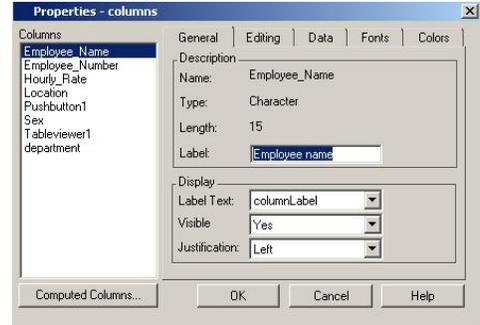
```
PushButton1:
  location="Orlando";
```

```
Return;
```

(Please note that the Viewer SCL code used “Location” and not “Location.text”, that is because “Location” is a column name within the Form Viewer and not frame object.) It is also necessary to right click on the Form Viewer, select “Form ->”, then “Display column window”, and then “Make Computed Column”. Drag it over the push button on the Form Viewer. This allows the PushButton1 labeled section in the model SCL to execute when the push button is selected.

#### Q16: How do I modify the attributes of a component within a Viewer.

**A16:** To modify the attributes of a component within the viewer, click on the columns attribute of the viewer then select the column (component) which you would like to modify.



#### Q17: I created a Form Viewer many days ago, since then I have spent a lot of time customizing its appearance. During my last session I clicked the SAS® close button to close my session (forgetting to close my AF build session first).

**SAS® terminated but did not ask me to SAVE before closing. When I re-entered I discovered that all my customizations ha\$ been lost. HELP!!!**

**A17:** VERY IMPORTANT!!! This is a reported problem in Release 8.1 and 8.2. ALWAYS close your Build frame screen before closing SAS® otherwise customizations to the Form Viewer are lost and the Viewer reverts back to its default layout. It is wise to periodically make a copy of the frame. This can be done at the Build Explorer Window by right clicking on the frame and selecting “Duplicate...”.

This issue is addressed by the following SAS® Note.

<http://www.sas.com/service/techsup/unotes/SN/003/003717.html>

#### Q18: How can I change the TAB order within a Form Viewer?

**A18:** Right click within the Form Viewer , next select Form -> and then select “Tab order”.

#### Q19: I created a combo box as was discussed in Question1, but when I run in TESTAF I find that the text is blank for the first observation but is displayed in subsequent observations. Why does it not show up?

**A19:** Check to see if the values in your items list for the combo box match the value for the column linked to the combo box in the first row. When the form viewer is first displayed SAS® does a \_refresh to all the components, just for the first row. When this is done SAS® checks to see if the value matches, if it doesn't, the value is not displayed. (Hint: also check capitalizations)

#### Q20: I want to add a button on the frame such that when I push the button the Form Viewer adds a new observation to the SAS® data set. How can I do this?

**A20:** Create a push button outside of the Form Viewer, then add the following code to the frame's SCL. It executes when the push button is pressed.

```
PushButtonADD:
    Sasdataset1._addRow();
return;
```

It is also useful to add an additional button to save the input once the user has finished (SAS® will save automatically if another row is added or if you exit out of add mode; a SAVE push button would have the following code in the frame.

```
Sasdataset1._commitNewRow();
```

Please note that the methods \_addrow and \_commitNewRow are applied to the sas data set model and not to the Form Viewer. If you wanted to CANCEL the input and return to previous values, use the following method.

```
Sasdataset1._rereadCurrentRow();
```

#### Q21: How can I get the Form Viewer to advance to the next row or go back one row?

**A21:** If you want to go directly to a specific row number (example: observation 500) issue the following methods to the Form Viewer:

```
formviewer1._gotoRowNumber(500);
```

Substitute:     -1 for first row  
                   -2 for previous row  
                   -3 for next row  
                   -4 for last row

The `_vscroll` method is also supported, see SAS® Help.

(Note: if you place your cursor within the Form Viewer while in run mode, you can use the PgUp and PgDn keys – even faster)

**Q22: I'm still unable to move to the last row in a Form Viewer from the frame's INIT section. I've tried `_gotoRowNumber` but it didn't work. How do I do it?**

**A22:** To circumvent the problem try this in the frame's SCL.

```
INIT:
    _frame._refresh();
    formviewer1.gotoRowNumber(-4);
return;
```

This last item is discussed in the following SAS® Note.

<http://www.sas.com/service/techsup/unotes/SN/005/005450.html>

**Q23: I want to create a PMENU allowing me to select a department from the menu. Once I have made my selection I would like to subset my Form Viewer by this value. How can this be done.**

**A23:** You could use code below to generate a “pmenu”:

```
proc pmenu c=sasuser.sugi27;
    MENU INITIAL;
        item 'Exit' selection=exit;
        item 'Subset..' menu=sub;
        selection exit 'end;';
    MENU SUB;
        item 'By Department' menu=departm;
        item 'CLEAR SUBSET' selection=csub;
            selection csub 'WCLEAR';
    MENU departm;
        item "Marketing" selection=marksel;
        item "Accounting" selection=accsel;
        selection marksel 'MARKETSEL';
        selection accsel 'ACCOUNTSEL';
run;
quit;
```

Next the frame “pmenuEntry” attribute must be modified so that the pmenu will be displayed at the top of the screen. In addition the frame's SCL must be modified to capture the commands such as “MARKETSEL” which will be sent by the pmenu to the frame's command line.

```
MAIN:
    control allcmds ;
    word=word(1,'u');
    select (word);
        when ('MARKETSEL') do;
            sasdataset1.where="department='MARKETING' ";
            call nextcmd();
            end;
        when ('WCLEAR' ) do; call nextcmd() ;
            sasdataset1.where=' ';
            formviewer1._VSCROLL('max',+1);
            end;
```

```

        otherwise
    if word ne _blank_ then
        sasdataset1._execcmd();
    end;
return;

```

This frame SCL code detects the word sent to the frame's command line; if the word is "MARKETSEL" it subsets the sas data model, if "WCLEAR" the where clause is undone. The \_execcmd() method passes other words intercepted on the command line, for example typing "3" would take you directly to the 3<sup>rd</sup> observation.

**Q24: Can I generate a title at the top of the frame showing which observation I'm on?**

**A24:** Yes, the following could be added to the frame SCL.

```

MAIN:
    sasdataset1._getRowNumber(currow);
    if (currow=0) then _frame_.title='No Rows';
return;

```

Add this to the Form Viewer model scl.

```

dcl list infolst={}, list datalst={}, object frameid;

DFINIT:
    frameid=_viewer_.frameid;
return;

INIT:
    _self_.getRowStatus(infolst);
    new=getnitemc(infolst,'new');
    if new='Y' then frameid.title='Pending Row';
else do;
    _self_.getRowNumber(currow);
    _self_.getMaxRow(maxrow);
    if maxrow > 0 then do;
        _self_.getDatasetAttributes(datalst);
        numdel=getnitemn(datalst,'number_of_deleted_rows');
        if numdel > 0 then maxrow=numdel+maxrow;
        frameid.title='Row '||currow||' of '||maxrow||' rows';
    end;
    else do; _self_.getMaxRow(maxrow,'y');
        frameid.title='Subset mode '||maxrow||' rows found';
    end; end;
return;

DFTERM: rc=dellist(infolst);
        rc=dellist(datalst);
return;

```

Using this SCL will generate a different title when the data set is being subset. The value of MAXROW < 1 when the data is being subset.

**Q25: How can I change order of the columns within a Table Viewer? How can I hide a column?**

**A25:** Right click within the Table Viewer , select Properties and then select "column order". Notice, you can also hide and unhide columns here.

**Q26: How can I generate a combo box for one of the columns within a Table Viewer? What different methods are there to fill the box?**

**A26:** It's not difficult. Right click and select properties of the SAS® data set model used for the Table Viewer. Go to the columns attribute. Click the DATA tab. If you wanted a “DropDownComboBox” that could be selected instead of TextEntry. In ValuesList you could also chose “Entered list of values” to add values directly, or select “unique column values” to pull values from a data set. You can also add values from “SCL or SLIST entry”.

If you have a SAS® Data Set Model that you have attached to a Table Viewer control within a SAS®/AF FRAME entry, you can use a separate SCL entry to build an SCL list of the values you wish to display via the cell controls drop-down list, drop-down combo box or spin box. To use this feature, programmatically define the valuesList attribute as follows: (in this case Sasdataset1 is a linked Table Viewer)

```
Sasdataset1.columns{colnum}.valuesList='\lib.cat.buildSCLList.scl';
```

where lib.cat.buildSCLList.scl is the four-level name of the SCL entry that contains the logic to create the SCL list.

At minimum, this SCL entry needs to contain the following statements:

```
entry optional=listOfValues:list objectID:object;

INIT:
  dcl list myValueList={};
  /* logic to populate the SCL list myValueList with values to
  display via a cell control */
return;

TERM:
  listOfValues=copylist(myValueList, '', listOfValues);
  if myValueList then myValueList=dellist(myValueList);
return;
```

In the above SCL program, the SCL list myValueList would be populated with the values that you wish to display via the cell control that you have designated for your column.

#### **Q27: How do I link a Table Viewer that is nested within a Form Viewer?**

**A27:** This is actually very easy to do. First, make sure that you click within the first Form Viewer to activate it, then bring up the “Display columns window” discussed in question #1, next, click on the key column of the Form Viewer and drag it the Table Viewer. The drop can occur anywhere on the Table Viewer. In order for this to work, you must also ensure that the “keyColumn” attribute, in the SAS® data set model (sasdataset2), is set to the key variable. The keyColumn only needs to be set in the model for the Table Viewer.

Using this approach it is not necessary to have the SAS® data sets indexed by the key column – however, doing so might improve performance in large data sets.

#### **Q28: I've tried linking a Form Viewer that is nested within another Form Viewer the same way as I did for Question #27 but it didn't work. What did I do wrong.**

**A28:** There is a slightly different procedure for nesting one Form Viewer within another. You also must click in the main Form Viewer (formviewer1) to activate it. To complete the linkage you must type the following command on the command line: “LINK FORMVIEWER2 EMPLOYEE\_NUMBER” and press enter– here formviewer2 is the name of the nested Viewer and employee\_number is the key column of both the models.

#### **Q29: In a master/detail application, I want to create a pushbutton on my Form Viewer such that when it is pushed it would ADD a record to the Table Viewer matching the key value (eg. Employee number) currently in the Form Viewer?**

**A29:** Insert SCL code similar to what is below into the SCL for your Form Viewer. The “DFINIT” section identifies the Table Viewer to the Form Viewer. Once the ADD pushbutton is pressed the value of the key column is updated and the addRow method is executed.

```
DFINIT: dcl object tableid modelid;
  _viewer_._getComponentID('tableviewer1',tableid);
  modelid=tableid.modelid;
return;

INIT:
  pushbutton1='ADD';
```

```

return;

PUSHBUTTON1:
  modelid._getColumnNumber('Employee_number', colnum);
  modelid.columns{colnum}.initialValue.numericValue=Employee_number;
  modelid._addRow();
return;

```

Setting pushbutton1 to “ADD” makes the word “ADD” visible on the button. This last item is discussed in the following SAS® Note.

<http://www.sas.com/service/techsup/unotes/SN/000/000235.html>

### Additional Answers from SAS on SAS/AF problems.

**Q30 The value in a combo box contained in my Form Viewer displays incorrectly I the combo box is active and I use Pageup/Pagedown keys to scroll rows within my Form Viewer.**

**A30:** This problem is discussed and remedies are supplied within the following SAS® Note:

<http://www.sas.com/service/techsup/unotes/SN/004/004155.html>

**Q31 If I set the visible attribute on a column within my Form Viewer I lose the lock on the row within that Viewer.**

**A31:** This problem is discussed within the following SAS® Note:

<http://www.sas.com/service/techsup/unotes/SN/004/004313.html>

**Q32 I have a combo box using formatted values within my Form Viewer but the values are not displayed properly. How can I fix this problem?**

**A32:** This problem is discussed within the following SAS® Note:

<http://www.sas.com/service/techsup/unotes/SN/002/002524.html>

**Q33 Sometimes I get anabend after modifying a my Master/Detail application and then click on a Version 8 control? What's happening and how can it be fixed?**

**A33:** This problem is discussed within the following SAS® Note:

<http://www.sas.com/service/techsup/unotes/SN/005/005606.html>

**Q34 When I type a value into a text pad control the resultant display always gets capitalized. I've set uppercase to “NO” in the columns attribute window but it does not have any effect. Why does this happen?**

**A34:** The “caps” value in the Form Viewer control object gets set by the value that is encountered in the first row of the table. To circumvent this problem, the control with which you are having the problem can be set programmatically such that the results are displayed in lower case. This problem is addressed in the following SAS® Note.

<http://www.sas.com/service/techsup/unotes/SN/003/003925.html>

### ACKNOWLEDGMENTS

I wish to express my sincere thanks to the employees at SAS® for the tremendous assistance they have rendered me. Special thanks go to Ann Rockett and Lynn Curley from SAS Institute. Many of the ideas and concepts expressed in this paper can be attributed to their hard work.

### CONTACT INFORMATION

Your comments and questions are valued and encouraged.

Contact : Bernd E. Imken

Chief, Information Systems Division

Patented Medicine Prices Review Board, Government of Canada

333 Laurier Avenue West, Ottawa, Ontario, Canada . K1P 1C1

Phone: (613) 952-3312

Fax: (613) 952-7626

Email: [bimken@pmprb-cepmb.gc.ca](mailto:bimken@pmprb-cepmb.gc.ca)

Web: [www.pmprb-cepmb.gc.ca](http://www.pmprb-cepmb.gc.ca)