

Paper 46-27

Developing Master/Detail Applications

Using Form & Table Viewers

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

ABSTRACT

SAS® has advanced remarkably with ODS in providing quality output, now Form & Table Viewers provide this quality at the screen level. This paper demonstrates how to develop applications using SAS/AF® Release 8.2. It explains how FORM and TABLE VIEWERS are developed, modified and eventually linked together to produce a Master/Detail application. The “Frequently Asked Questions” approach focuses on typical difficulties encountered during applications development.

INTRODUCTION

SAS® 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® added VIEWTABLE. Although easy to use, these procedures lacked the high level of customization demanded by the developers. 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.

CREATING FORM & TABLE VIEWERS

I described the process of creating FORM and TABLE VIEWERS, in detail, in a paper entitled “A SAS® Based Correspondence Management System” published with Sugi26. It is available at:

<http://www2.sas.com/proceedings/sugi26/p041-26.pdf>

This Sugi26 paper also describes how to link viewers together. In summary, creating a Form Viewer is really quite simple.

1. drag the Form Viewer Control Component onto a frame
2. drag a SAS Data Set Model Component over the blank Form Viewer which was created in step 1
3. Right click on the Form Viewer and select Properties, then change the Table Attribute for Sasdataset1 (the default name to the model) to point to the data set you want to display in the Form Viewer.

You can create a Table Viewer using a similar approach. In addition, you can also nest multiple Viewers on a single Frame. For example, you can create a Table Viewer within a Form Viewer.

Each Form and Table Viewer also has its own SCL code, identified by the attribute “ModelSCLEntry”. In addition, the VIEWER SCL code supports “DFINIT” and “DFTERM” sections allowing you to control a SAS® data set prior to the first record being read or after processing the TERM of the last record. This is useful (for example) if you open an external data set as you entered the Viewer and close it only when you exit the Viewer. (Hint: `rc=close(-999);` closes all open files)

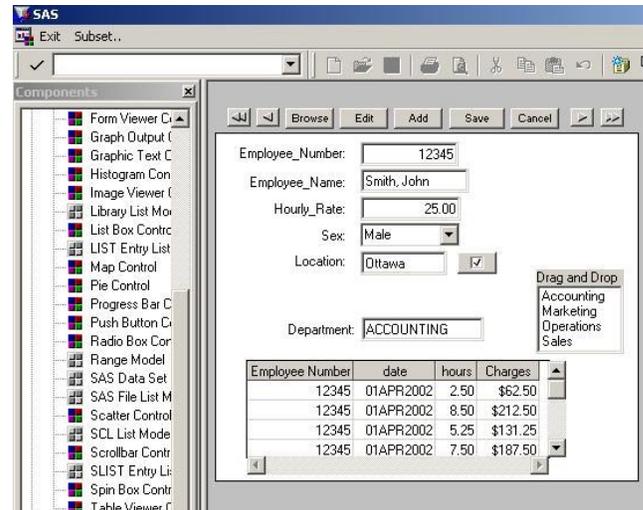


Figure 1. Display of Form Viewer with a linked Table Viewer

Figure 1 shows a simple example of a Master/Detail relationship between a Form Viewer for an employee and a Table Viewer displaying hours worked linked using a one to many relationship. To link a “employee” Form Viewer to a “work_hours” Table Viewer:

1. right click on the Form Viewer then select “Form ->”, then select “Display column window”
2. click on the “Employee_Number” in the “Display Column Window” and drag it over the “Employee_Number” column of the Table Viewer
3. set the “keyColumn” attribute to “Employee_Number” for the SAS data set model associated with the Table Viewer(Sasdataset2 - note: for this paper I will retain their default names)

I discussed alternate methods of linking Form and Table Viewers in my Sugi26 paper. I also give SAS® code to link Viewers, which are not nested, but still exist on the same frame.

CUSTOMIZING FORM & TABLE VIEWERS, CREATING MASTER/DETAIL APPLICATIONS

You can customize Form & Table Viewers by adding push buttons, graphics, combo boxes etc. You can now also create Master/Detail relationships between Viewers. There are, however, things which every developer needs to know to get things working properly. Amongst the most common questions in doing such development are:

Q1: 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?

A1: 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.

Q2: How can I determine if a screen object is linked to a data set variable?

A2: 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.

Q3: 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?

A3: 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.



Figure 2. Modifying the ".columns" attribute

Q4: When I originally created a Form Viewer the text label beside the text entry "Employee_Name" was created. I right clicked on this text label to change the label attribute to "Employee #"; everything at first looked right but when I went into TESTAF mode the label reverted back to "Employee_Number". What's happening?

A4: This problem existed in Release 8.1 but was fixed for Release 8.2. It was explained in the following SAS Note. <http://www.sas.com/service/techsup/unotes/SN/005/005509.html>

The columns attribute within the model also allows you to set format, informat, maximum, minimum, required, protected etc.

Q5: 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 had been lost. HELP!!!

A5: VERY IMPORTANT!!! This is a reported problem in Version 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..."

Q6: 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?

A6: 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)

Q7: 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?

A7: 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();
```

Q8: How can I get the Form Viewer to advance to the next row or go back one row?

A8: 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 formviewer while in run mode, you can use the PgUp and PgDn keys – even faster)

Q9: 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?

A9: To circumvent the problem try this in the frame's SCL.

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

Q10: When I create a Table Viewer within a Form Viewer and both are set to 'BROWSE' the scroll bars on the Table Viewer disappear. How can I enable scrolling?

A10: When you display a data set in browse mode within the Form Viewer, the controls within the Table Viewer are not enabled (ie. Protected). Thus the scroll bars are not enabled. To circumvent the problem, open the data set displayed in the Form Viewer with "editmode" attribute set to "rowLevelEdit" and protect all the columns except the Table Viewer (TABLEVIEWER1) This can be done using the frame SCL code below.

```
dcl list display_list={}, char(8) colname;
init:
sasdataset1._getDisplayedColumns(display_list)
;
do i=1 to listlen(display_list);
  colname=getitemc(display_list,i);
if colname in ('EMPLOYEE_NAME' 'EMPLOYEE_NUM')
  then do;
sasdataset1._getColumnNumber(colname,colnum);
sasdataset1.columns{colnum}.protected='YES';
```

```

end;
end;
if display_list then
display_list=dellist(display_list);

```

Once this code has executed, you will be able to scroll the Table Viewer but you will be unable to edit any values displayed in the Form Viewer (in this example, for brevity, only two variables were protected). The values displayed within the Table Viewer are also protected because the model linked to that Viewer still has the "editmode" attribute set to "Browse". Using this approach is also very useful if you wanted to programmatically control objects within the Form Viewer. For example, if had a push button (PBut) on the Form Viewer and you wanted to hide it you could issue code such as:

```

sasdataset1._getColumnNumber('PBut',colnum);
sasdataset1.columns{colnum}.visible='NO';

```

Q11: 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.

A11: 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 3rd observation.

Q12: Can I use drag and drop with Form & Table Viewers ?

A12: With Form Viewers it's easy. A drag site can exist either outside or within a Form Viewer. A drag site such as a list box must have "dragEnabled" set to YES. In addition, you would right click on the text box (drop site) within the Form Viewer, and set the "dropEnabled" attribute to YES. However, with Table Viewers the process is not quite so simple, you have to define to the model exactly what column the value is to be assigned to. Because the drop information needs to be initialized at the time

the model in instantiated, you will need to create a subclass of the SAS Data Set Model Class. Since space for this paper is limited, the methods required and how to apply them to generate a new sub class can not be detailed. Please contact me if you have this requirement. Copies of the required code and instructions will be available at the Sugi27 conference.

Q13: Can I generate a title at the top of the frame showing which observation I'm on?.

A13: 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_r
ows');
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.

Q14: 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?

A14: 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!

Q15: 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?

A15: 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 to 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.

Q16: 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?

A16: 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;
  /* note:above 2 lines is really 1 line
  - no spaces*/
  modelid._addRow();
return;
```

Setting pushbutton1 to “ADD” makes the word “ADD” visible on the button.

Q17: When I nest 1 Form Viewer within another Form Viewer I find that I cannot move objects around on the second viewer. What's wrong?

A17: The developers looked into this at that time and found that it was only a problem on PC platforms. It had to do with a conflict with cursor tracking and drag and drop status and they couldn't

find a solution that wouldn't cause them to turn off some functionality that was needed so they were not able to change the behavior.

A circumvention that might help you position your columns a little better than when you cut and paste is to select the column, then select the Layout pmenu item and then the Move pmenu item.

This gives you the outline of the column that you can move around and position as you would like. This is much easier to position than when you do a paste action and aren't sure exactly where your pasted column is going to wind up.

Q18: How do I adjust the column width within a nested Table Viewer?

A18: If the Table Viewer is nested within a Form Viewer, you will find that you cannot directly adjust the column widths by dragging the label border as can be done when a Table Viewer is not nested. One trick is to cut the Table Viewer and paste it outside of the Form Viewer. Now adjust your column widths. Once completed, cut the adjusted Table Viewer and paste it back into the Form Viewer. (Note: it is wise to take a duplicate copy of your entire frame first, in case you do something incorrectly so as not to lose your original.)

Q19: How can I change the TAB order within a Form Viewer?

A19: Right click within the Form Viewer, next select Form -> and then select “Tab order”.

Q20: How can I change order of the columns within a Table Viewer? How can I hide a column?

A20: Right click within the Table Viewer, select Properties and then select “column order”. Notice, you can also hide and unhide columns here.

Q21: How can I create a computed column within a Form Viewer or Table Viewer?

A21: 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.

Q22: How can I wrap text in a text object within a Form Viewer?

A22: In the example shown below a “synopsis” column is displayed within a text pad control. Within the original SAS® data step the length of the synopsis variable was set by the following SAS® statement:

```
length synopsis $ 500;
```

SAS® now can support very long character variables up to 32,767 characters in length (note – prior to Version 8 a character variable was restricted to 200 characters.) If you are using long text strings of this type it may also be useful for you to specify the data set option “compress=YES”. In order to achieve the data wrapping, replace the original text entry control with a text pad control and set the scrollbar to “vertical only”. Remember to re-link the synopsis column with the text pad control as described in Q1. (Note: the text pad only wraps on PC platforms, not currently on Unix platforms.)

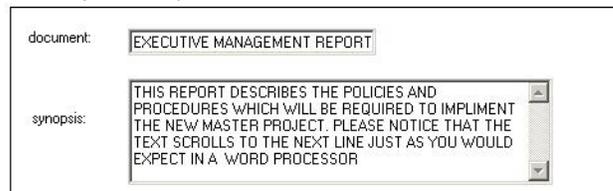


Figure 3 Creating a wrapped text pad column in the Form Viewer

Q23: How can I wrap text in a column within a Table Viewer?

A23: In order to do this, you have to set the “wrapDataText” attribute to “YES” in the SAS data set model attached to the Table Viewer. The process however is not as intuitive as only setting the attribute. If :

```
length synopsis $ 500;
```

then the column which would be generated by the Table Viewer when it is first set up would also be of this same width (ie. 500 characters) even though “wrap” were specified. The problem is to reduce the width of the column in the first place. This is not so difficult if you were moving from a column width of 60 characters to 30, because the initial 60 columns could have been displayed on the first Table Viewer, you could then have simply clicked on the right hand side of the label border and dragged it to the left to reduce the column width. If the initial size of the character column were very large, it is easier to first create a Table viewer from a data set where the synopsis column was say 30 characters. After the Table Viewer was created, go back to another data step and re-define the length to be 500; Next go to the “column” attributes and set the format and informat from \$30. to \$500. The usage of the column, however, still acts different from the wrapping in a text pad object. The data input is still done on 1 streaming line – it is the output text only which is wrapped as soon as you move your cursor to another field.

Starting in Release 8.2, we now have a programmatic way to set column width by applying the the following methods to the SAS Data Set Model:

```
_setColumnWidth(colname,width);
_getColumnWidth(colname,width);
```

Q24: Can I freeze a column in a specific location within a nested Table Viewer?

A24: Set attribute for “heldColumns for the Table Viewer control .

| | document | synopsis |
|---|-----------------------------|---|
| 1 | EXECUTIVE MANAGEMENT REPORT | THIS REPORT DESCRIBES THE POLICIES AND PROCEDURES WHICH WILL BE REQUIRED TO IMPLEMENT THE NEW MASTER PROJECT. PLEASE NOTICE THAT THE TEXT SCROLLS ON LINE LINE UNTIL THE ENTER BUTTON IT HIT. ONCE THE YOU MOVE YOU CURSOR TO ANOHTHE CELL THE FORMAT WILL TAKE PLACE |
| 2 | NEW REPORT | ONLY LIMITED TEXT FOR THIS ONE. |

Figure 4 Creating wrapped text for a column in a text viewer.

Q25: How do I link a Table Viewer that is nested within a Form Viewer?

A25: 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.

Q26: I've tried linking a Form Viewer that is nested within another Form Viewer the same way as I did for Question #25 but it didn't work. What did I do wrong.

A26: 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.

Q27: I find it confusing sometimes determining if a given method should be applied to the Viewer or to the SAS® Data set Model. Can you help me?

A27: I found it most useful to look this up under the SAS® Help, under the SAS/AF Component References you will see both Visual Components and Models. Methods that apply to the Form and Table Viewers are listed respectively within the Visual Component under the Viewers name. The SAS data set model and other models such as the variable values list model are covered under Models. SAS/AF® Component References are found by clicking the “CONTENTS” tab, going to the Help on SAS® Software products and then selecting SAS/AF Software.

Q28: 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?

A28: 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.

Q29: I found that linking two separate Form Viewers existing within one frame did not work the same way as in the last two questions. How can two separate Form Viewers be linked?

A29: There is a slightly different procedure for linking controls which are not nested. SAS® generally gives examples of Master/Detail relationships which are nested. Occasionally however, an application design requires 2 separate controls. As of SAS® Version 8.2 linking one Form Viewer to another, or linking one Form Viewer to a separate Table Viewer can be accomplished in a number of different ways. You could use the _setKey method. Using one approach, the code exists solely within the first Form Viewer. In the second some SCL code exists at the frame level and the rest is within the Form Viewer SCL. The code can be obtained in my Sugi26 paper on page 5 and can be obtained at: <http://www2.sas.com/proceedings/sugi26/p041-26.pdf> The example code shows both approaches. The example shows “formviewer1” being linked externally to “formviewer2”, the key variable in the example is “mailno”. For this to work properly please make sure that each data set model has the “keyColumn” attribute set to “mailno” and that each data set is also indexed by “mailno” as the primary key.

Q30 I have a Table Viewer control within a Form Viewer control making up a master/detail application. When I lock a row in the table viewer and then scroll the form viewer to the next row, the form viewer and table viewer will no longer scroll in sync. I receive a program halt in the _setKey method with the following error; “The setting of object 'yourtableviewer' to the value of column 'yourkeycolumn' failed.”

A30: This error only occurs if the data set for the table viewer is indexed. If the table viewer is in table level edit, no errors are produced but the form viewer and table viewer are no longer in sync and you cannot END from the frame. To circumvent the problem, remove the index from the data set attached to the table viewer.

A Technical Support hot fix for Release 8.2 TSLEVEL 2M0 for this problem is available at: <http://www.sas.com/techsup/download/hotfix/v82/base/82ba27/82ba27.html>

Q31: I have a Combo box within a Form Viewer, linked to a numeric column. Why do I get the following error message? ERROR: The setting of object 'combobox1' to the value of column 'XXX' failed.

A31: The selectedItem attribute for the Combo box expects a character value. A work around is provided by SAS at:
<http://www.sas.com/service/techsup/unotes/SN/001/001115.html>

CONSIDERATIONS FOR USING NESTED VIEWERS VS. EXTERNAL VIEWERS

When designing a new application developers frequently encounters tradeoffs between simplicity in design versus performance. There are obviously pros and cons in determining if viewers should be nested or external from one another when designing a Master/Detail relationship model. Generally, at the present, using nested structures is much easier to implement than external linkages. Linking with _setKey is not very good for combining other values unless everything is using an EQ condition and the variables are both in a composite index (for example: doing subsets over two Viewers). Another approach would be to use only WHERE clauses and not use _setKey. The advantage of _setKey is that you know it is going to use the index. With a WHERE clause, SAS[®] decides whether to use the index. However, in Version 8, we now have the IDXWHERE=YES data set option that forces SAS[®] to use the index, even if a sequential pass of the data is more efficient. Now you could set the WHERE clause to do the linkage between the Form Viewer and an external Table Viewer.):

```
modelidDemo.where='employee_number='  
||quote(employee_number);
```

CONCLUSION

I have had considerable experience developing Form and Table Viewers during the last two years. The process now appears simple as compared to when I started. Throughout this paper I have tried to give you the questions which I have encountered during development.

ACKNOWLEDGMENTS

I wish to express my sincere thanks to the employees at SAS[®] for the tremendous assistance they have rendered me. Especially, Ann Rockett and Lynn Curley. Frequently they have pitched in to help me when I became stuck on an issue. Many of the ideas and concepts expressed in this paper can be attributed to their hard work. Hopefully, this paper will assist others in a similar fashion.

CONTACT INFORMATION

Your comments and questions are valued and encouraged.

Contact : Bernd E. Imken

Patented Medicine Prices Review Board

333 Laurier Avenue West

Ottawa, Ontario, Canada . K1P 1C1

Phone: (613) 952-3312

Fax: (613) 952-7626

Email: bimken@pmprb-cepmb.gc.ca

Web: www.pmprb-cepmb.gc.ca