

An Example of Using SAS/AF® DATA TABLE Class In Reviewing Clinical SAS® Data Sets

Irene Zhao, Allergan, Irvine, California

Abstract

Using SAS/AF as part of the quality assurance process in reviewing SAS data sets during a statistical analysis provides a quick solution to resolving data queries. A Graphical User Interactive (GUI) module based on a Data Table Object provides a convenient way to examine a SAS data set by pointing and clicking. The Data Table Class in SAS/AF with Screen Control Language (SCL) programming is very powerful in manipulating the way to browse SAS data sets. The example module called “Data Display” allows the user to browse data sets located on local disks or remote platforms such as Unix. The user can choose a data set to be viewed from a specified data library. The chosen data set is displayed by Data Table Object in a tabular format with multiple rows, in full size or a drill down subset by a value of a variable. This module consists of the following widgets: DATA TABLE, CONTROL, RADIO BOX, LIST BOX, and a few PUSH BUTTONS. Each of the widgets is supported by a labeled SCL programming section. It requires software SAS 6.12, SAS/AF, and SAS/CONNECT to run this module.

Introduction

Graphical User Interactive (GUI) module by using Data Table Class provides an efficient way to browse the SAS data set by pointing and clicking. It allows the user to browse data sets located on local disks or remote platforms such as Unix. The user can choose a data set to be viewed from a specified data library. The chosen data set can be displayed in full set or a drill down subset by a value of a variable. Data Table Object displays the selected data set in a tabular format with multiple rows: each row represents a record and each column represents a variable. It also supports “drag and drop” for users to move columns within the display table. For viewing more columns or more rows, the user can use the horizontal scrollbar or vertical scrollbar

respectively. For Data Display frame entry, see Figure 1.

Data Library Selection

Radio Box Object, labeled by “Data Directory” in Figure 1, with two items named “Local” and “Unix” provides data library selection. It is required to specify a library reference for the directory where the frame entry is stored before building the frame entry. In this example, the following *LIBNAME* statement specifies the library reference:

```
libname app "c:\app";
```

If data sets are stored on a remote platform, for example on Unix, then another *LIBNAME* statement with an option of “*server=unix*” is necessary to specify a library reference named “*rmtdata*” for the directory on Unix:

```
libname rmtdata "/unix/project/datasets"  
server=unix;
```

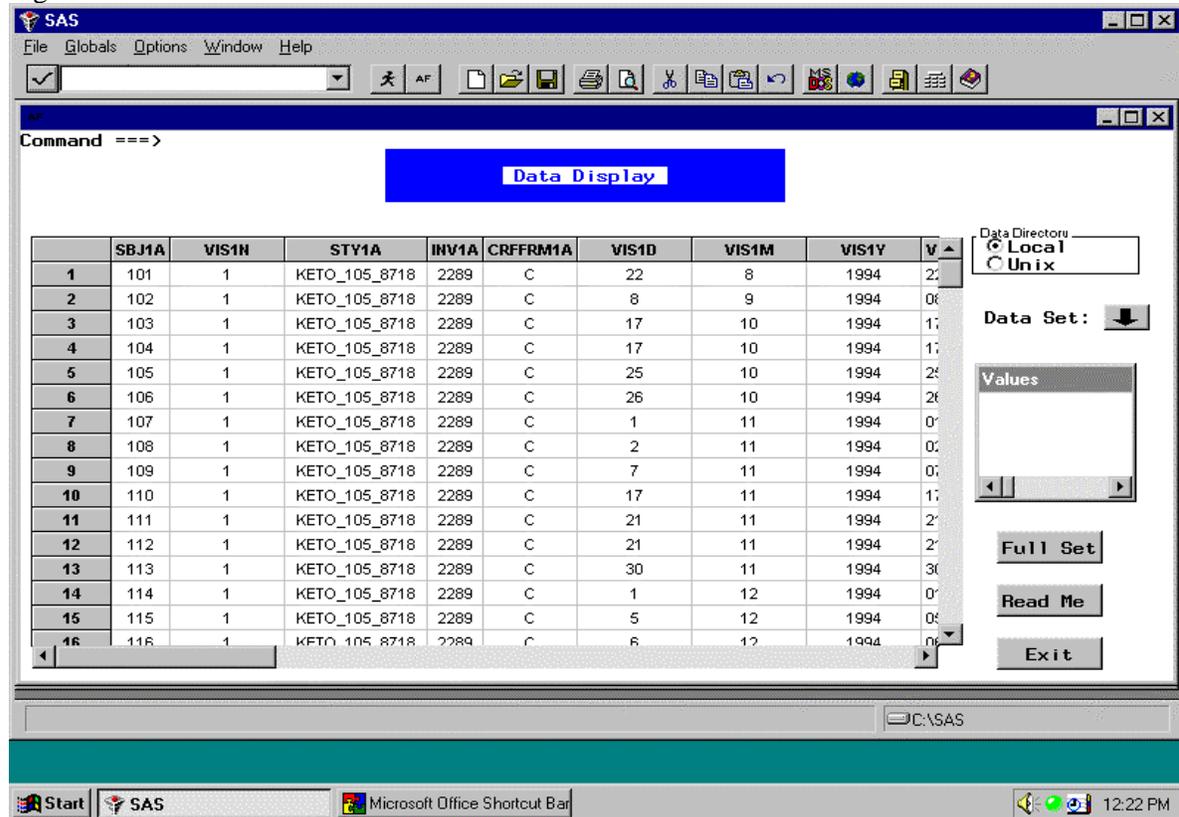
A labeled section of SCL entry for the radio box is as follows:

```
DATALIB:  
select (datalib);  
when ('LOCAL') dir='APP';  
when ('UNIX') dir='RMTDATA';  
end;  
return;
```

Data Set Selection

Control Object, the Arrow button in Figure 1, is designed to open the window, which lists the name of data sets in the specified data library. See Figure 2. The user can select any data set from the list. After a data set is selected, the Data Table object displays the selected data set in full size

Figure 1:



immediately. The following labeled SCL section is for the Control object:

CONTROL:

```

dolist=makelist();
rc=curlist(dolist);
dsselect=dirlist(dir,'data',1,'Y');
call
send(tabid,'_set_dataset_',dsselect,'browse','record
');
return;

```

Data Display

The selected data set is displayed by Data Table Object, the table object in Figure 1, in a tabular format with multiple rows: each row represents a record and each column represents a variable. Data Table Class provides “drag and drop” technique for users to rearrange the columns. For doing so, press the mouse select button with the mouse pointer in the column label and the column will be highlighted. With the mouse pointer in the highlighted column, click on the column and drag it onto another column. When the mouse button is released, the column dragged displays to the right of the

destination column. Data Table Class also provides horizontal and vertical scrollbars to display more records or variables, which are beyond the table edges. For SAS/AF and SCL programming experienced users, it is possible to create new variables and add columns to the table by editing the Model SCL entry. Data Display module discussed in this paper is designed for users with or without knowledge about SAS/AF and SCL programming, therefore the Model SCL entry is not discussed here. In Figure 1, a live SAS data set is displayed through the following steps: click on “Unix” in the radio box; then click the arrow button to open the specified Unix directory remotely, see Figure 2; click on the data set, named “demog” in the list, and the selected data set, “rmtdata.demog”, is displayed in the table. The SCL labeled section for the Data Table Object as shown below is referenced by Solution at Work: Rapid Applications Development Examples.

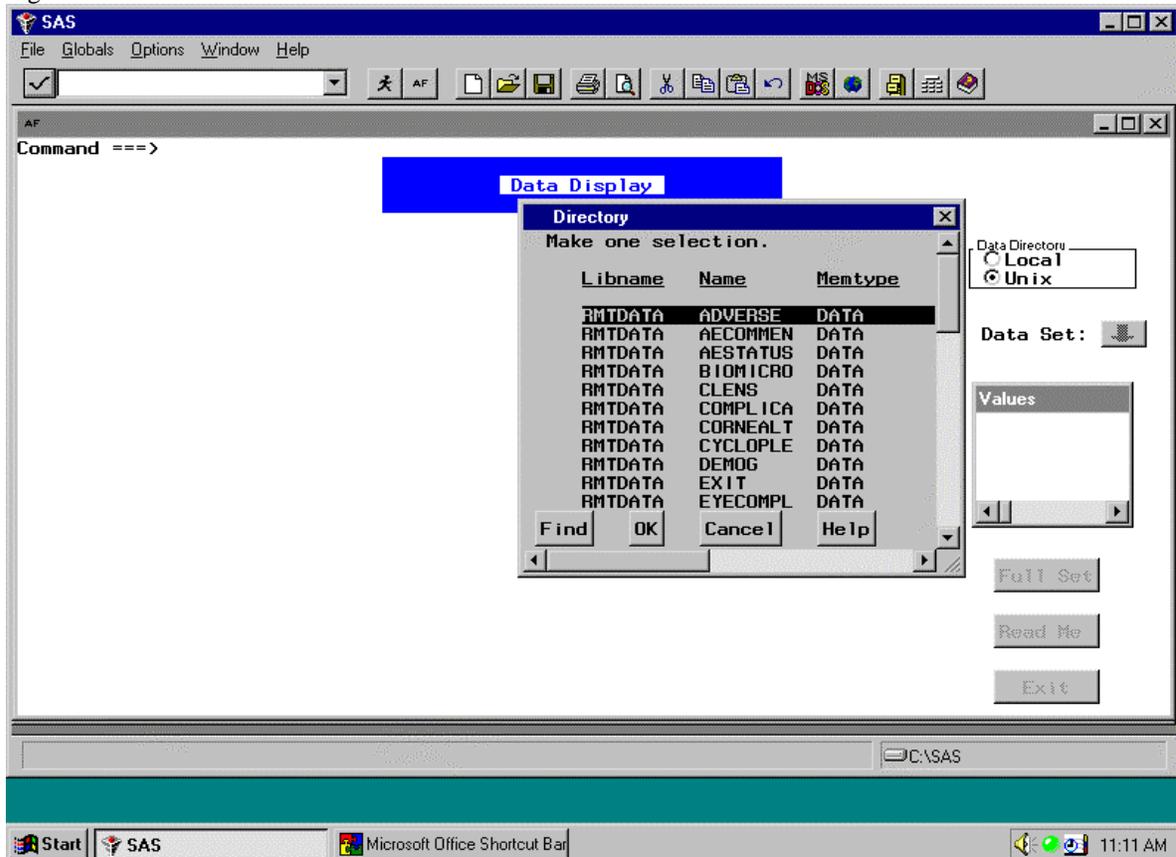
DATATAB:

```

call send(tabid,'_get_active_cell_',rowlist,collist);
colnum=getitemn(collist,1);

```

Figure 2:



```

call
send(tabid,'_get_displayed_column_name_',colnum
,colname);
call send(tabid,'_get_dataset_name_',dsname);
dsid=open(dsname);
nlevels=0;
rc=lvarlevel(dsid,colname,nlevels,vallist);
call send(boxid,'_update_');
return;

```

Subset Selection

By using LIST BOX Object, labeled by “Values” in figure 1, to list the values of the selected variable, the user should select one value from the list in the list box by clicking on it. As soon as the value is selected, the data displayed in the data table will be updated to a subset of the data set by the selected value of the variable. For example, if the user select variable SEX1F in the table, the list box will display values of 1 and 2; if the user clicks on value 2, then the data table

would display the records with sex1f=2 only. See Figure 3. At this point, the same data set can be re-subset by selecting another variable, for example SBJ1A, and a value of it. Sometimes data sets are created with a FORMAT statement in order to associate with a format catalog. In this case, some additional SCL programming about data formats is required. The labeled SCL section for this object is written for displaying the actual values of a data set without associating any formats. It is also referenced by Solution at Work: Rapid Application Development Examples.

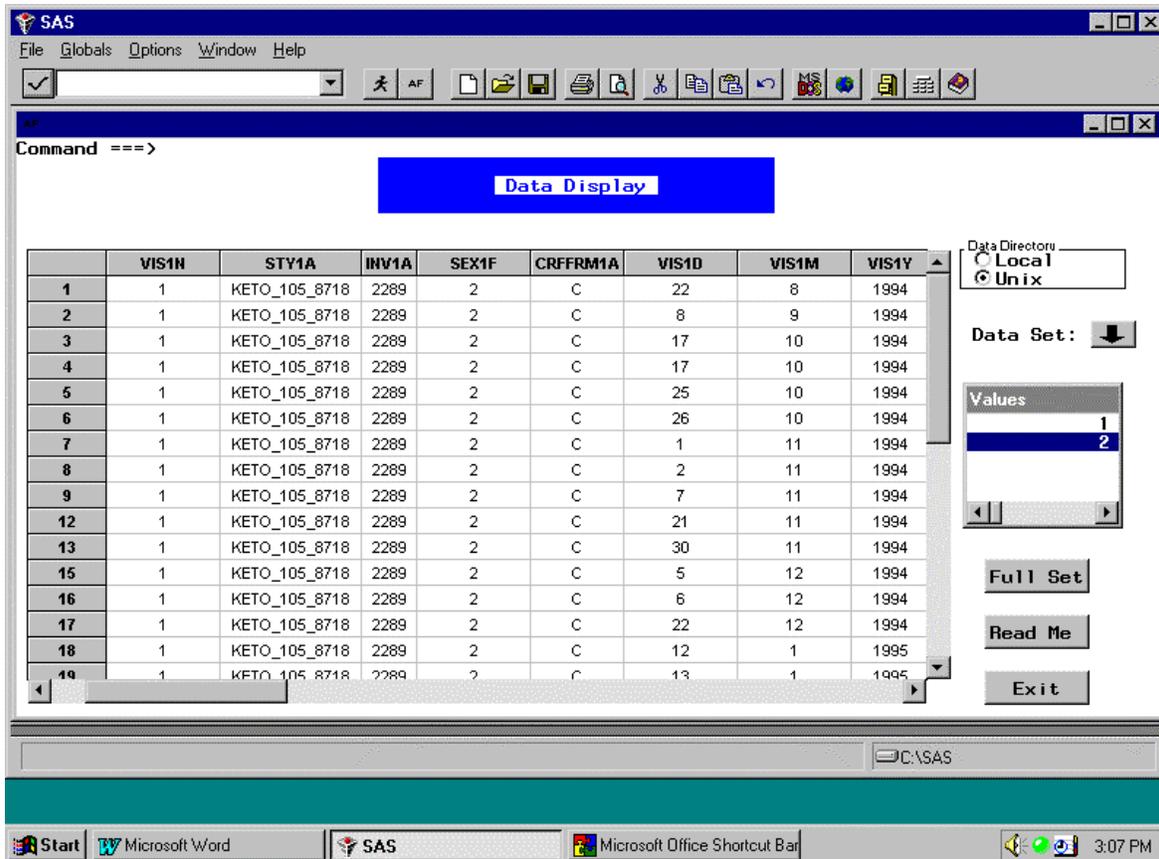
VALBOX:

```

call
send(boxid,'_get_last_sel_',rownum,issel,text
);
call
send(tabid,'_get_column_attribute_',colname
,'type',type);
if type='C' then

```

Figure 3.



```
rc=setitemc(wherelst,colname||'='||quote(text
),'Y');
else if type='N' then
rc=setitemc(wherelst,colname||'='||text,1,'Y')
;
call send(tabid,'_set_where_',wherelst);
return;
```

Returning to Full Data Set Display

The convenient feature for users that they can review the full data set again after sub-setting it. A push button “Full Set” is functioned to update the data table display from subset display to full data set display. It’s SCL labeled section is as follows:

```
FULLSET:
wherelst=clearlist(wherelst);
rc=setitemc(wherelst,'CLEAR',1,'Y');
call send(tabid,'_set_where_',wherelst);
return;
```

The SCL sections listed above are very basic source codes. To complete the SCL entry for

the frame entry, a length statement to define variables’ length, an INIT section, and a TERM section are necessary. It would be helpful to add a help button to describe the usage of the module. PUSH BUTTON named “Read Me” in Figure 1 is functioned to open a help file.

Conclusions

GUI module based on SAS/AF Data Table Class is a great tool for reviewing SAS data sets. It provides a convenient way to view SAS data sets locally or remotely by pointing and clicking. Data Table Object displays a SAS data set in a tabular format with multiple rows, in full size or a drill down subset by a value of a variable. Further more, this object allows users to move columns by “drag and drop” and view more columns and rows by horizontal and vertical scrollbars. The list box and the push button named “Full Set” allow users to toggle between subset and full set display, since the former object is function subset display and the later object is full set

display. In addition, the Data Display module can be built within any SAS/AF automated system, therefore users are able to review SAS data sets without leaving the system they are in. It is a convenient tool for experienced or inexperienced SAS users to get a quick solution to resolving data queries.

Acknowledgments

The following staff members of Allergan contributed to the preparation of this paper:

James Wang, Ph.D..
Russell Kondo

SAS is a registered trademark or trademark of SAS Institute Inc. in the USA and other countries.

® indicates USA registration.

Other product names are registered trademarks or trademarks of their respective companies.

Contact

Irene Zhao
Allergan
2525 Dupont Dr.
Irvine, CA 92623
Phone (714)246-6572
E-mail: irenez@allergan.com