Paper 072-2010

Next Generation DATA _NULL_ Report Writing using ODS OO Features: A Powerful Tool to Create Custom Tables

Alice Liqiong Li and Jacob Jen-Hao Cheng
Maryland Hospital Association - Quality Indicator Project, Elkridge MD

ABSTRACT

The ODS report writing interface feature is in preproduction for SAS 9.1.3® and later, presenting an object-oriented (OO) language fully integrating all capabilities of ODS and taking advantage of the rich programming features offered by DATA step. This paper provides an example of how this new technique can be used to create a complex table. Even users new to report writing will find this technique easy to use while offering a high degree of flexibility and control.

INTRODUCTION

The example tables presented in this paper are derived from a value-base purchasing (VBP) reporting project conducted by the Maryland Hospital Association’s Quality Indicator (QI) Project®. The QI Project® created customized pdf reports for hospitals calculating clinical and satisfaction performance scores and the resulting reimbursement. The first page of the report shows the hospital’s summary data in a table, including the hospital’s clinical measures and HCAHPS (Hospital Consumer Assessment of Healthcare Providers & Systems) measure performance scores for a baseline period and an assessment period, the overall VBP score, and the mean and 90th percentile of a comparison group.

Report features such as “traffic lighting” in cells, merging column cells and row cells, embedded images, and formatting cells based on another column value can be challenging to program with the PROC REPORT technique. This paper will demonstrate how SAS report writing using ODS OO features simplifies these tasks.

Note: For reasons of confidentiality and privacy, the measures are masked and reported as “measure X”. The actual report table contains more data than the examples shown in this paper.

OBJECT ORIENTED TECHNIQUES

The new ODS OO features in DATA _NULL_ report writing use the object oriented programming technique, the basic concepts of object-oriented are: object, class, instance, and method.

- Object is an entity with a bundle of attributes (elements) and actions (methods) to perform.
- Class is a blueprint or prototype from which objects are created.
- Instance is the actual object created at runtime. To create an instance, use NEW statement or DECALRE statement.
- Method means function in object-oriented world; a method can be called only by its own object. To call a method, use the object name follow by a dot, for example: an_object.method_name().

DATA _NULL_ REPORT WRITING USING ODS OO FEATURES

To create an instance of ODS report writing class (odsout); either of the following statements can be employed:

```
declare odsout obj();

obj = _new_ (odsout);
```

The object variable obj should be initialized only once for the life of the DATA step (initialization execution phase). Therefore, the variable obj should be instantiated on the first observation of the observation execution loop:

```
if _N_=1 the do;

        declare odsout obj();
        end;
```

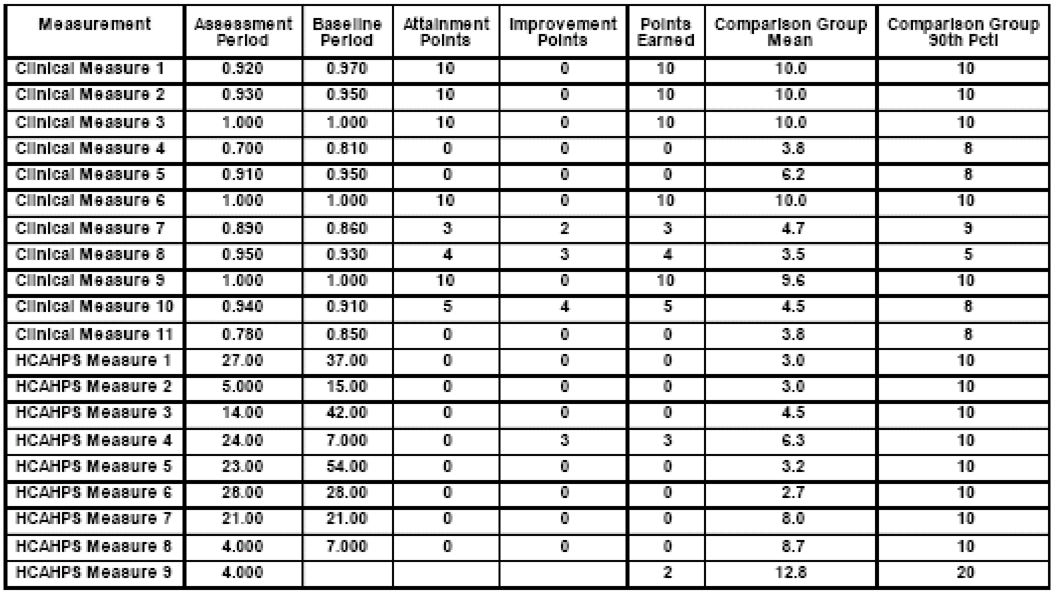
At the end of the DATA step (termination execution phase), the instance obj should be removed to allow any necessary termination processing with the object and free memory allocation. Use end=eof to detect if the execution reaches the last observation of the data set.

```
        data _null_;
        set sumtbl_data(end=eof);
        if eof = 1 then do;
                obj.delete();
        end;
        run;
```

During the observation execution loop, the power of DATA step functionality can be combined with the ODS report writing object methods to create customized reports.

EXAMPLE 1: AS SIMPLE AS A LIST

Create a report that simply lists the numbers in a table as shown below:



Measurement	Assessment Period	Baseline Period	Attainment Points	Improvement Points	Points Earned	Comparison Group Mean	Comparison Group 90th Pctl
Clinical Measure 1	0.920	0.970	10	0	10	10.0	10
Clinical Measure 2	0.930	0.950	10	0	10	10.0	10
Clinical Measure 3	1.000	1.000	10	0	10	10.0	10
Clinical Measure 4	0.700	0.810	0	0	0	3.8	8
Clinical Measure 5	0.910	0.950	0	0	0	6.2	8
Clinical Measure 6	1.000	1.000	10	0	10	10.0	10
Clinical Measure 7	0.890	0.860	3	2	3	4.7	9
Clinical Measure 8	0.950	0.930	4	3	4	3.5	5
Clinical Measure 9	1.000	1.000	10	0	10	9.6	10
Clinical Measure 10	0.940	0.910	5	4	5	4.5	8
Clinical Measure 11	0.780	0.850	0	0	0	3.8	8
HCAHPS Measure 1	27.00	37.00	0	0	0	3.0	10
HCAHPS Measure 2	5.000	15.00	0	0	0	3.0	10
HCAHPS Measure 3	14.00	42.00	0	0	0	4.5	10
HCAHPS Measure 4	24.00	7.000	0	3	3	6.3	10
HCAHPS Measure 5	23.00	54.00	0	0	0	3.2	10
HCAHPS Measure 6	28.00	28.00	0	0	0	2.7	10
HCAHPS Measure 7	21.00	21.00	0	0	0	8.0	10
HCAHPS Measure 8	4.000	7.000	0	0	0	8.7	10
HCAHPS Measure 9	4.000				2	12.8	20

The code to generate the table above:

```
%let colfmt=font_face=arial font_style=roman font_size=0.2 cellpadding=0 cellspacing=0 font_weight=bold just=center;

%let datafmt=font_face=arial font_style=roman font_size=0.2 cellheight=0.15in cellpadding=0 cellspacing=0 just=center font_weight=bold;
```

Create macro variable colfmt for column format definition, datafmt for data format definition

```
options nodate nonumber center missing=' ' orientation=landscape;

ods noptitle; ods noproctitle;

ods pdf file="ex1.pdf" notoc;

title; footnote;

proc sort data=sumtbl_data; by measure_type ord; run;
```

Send output file to ex1.pdf file, set option to suppress date (nodate) and page number (nonumber), print space if numeric value is missing, and set page orientation; notoc option omit the generation of bookmark list.

```
data _null_;
set sumtbl_data end=eof;
if _N_ =1 then do;
        dcl odsout obj();
        obj.table_start(name: "VBP Report");
        obj.row_start();
        obj.format_cell(data: "Measurement", overrides: "&colfmt. cellheight=0.3in");
        obj.format_cell(data: "Assessment*Period", split: '*', overrides: "&colfmt. cellheight=0.3in");
        obj.format_cell(data: "Baseline*Period", split: '*',overrides: "&colfmt. cellheight=0.3in");
        obj.format_cell(data: "Attainment*Points", split: '*',overrides: "&colfmt. cellheight=0.3in");
        obj.format_cell(data: "Improvement*Points", split: '*', overrides: "&colfmt. cellheight=0.3in");
        obj.format_cell(data: "Points*Earned", split: '*', overrides: "&colfmt. cellheight=0.3in");
        obj.format_cell(data: "Comparison Group*Mean", split: '*', overrides: "&colfmt. cellheight=0.3in");
        obj.format_cell(data: "Comparison Group *90th Pctl", split: '*', overrides: "&colfmt. cellheight=0.3in");
        obj.row_end();
end;
```

If _N_=1, it is the initialization execution phase, tasks that are executed only once at the beginning are: start a table, start first row as column name. Note: use split: '*' to break the column name into two lines so the table can fit on one page.

```
obj.row_start();
obj.format_cell(data: trim(left(measure_name)), overrides: "&datafmt. just=left font_weight=bold");
obj.format_cell(data: put(curr_rate, 5.3), overrides: "&datafmt.");
obj.format_cell(data: put(base_rate, 5.3), overrides: "&datafmt.");
obj.format_cell(data: put(att_score, 2.), overrides: "&datafmt.");
obj.format_cell(data: put(imp_score, 2.), overrides: "&datafmt.");
obj.format_cell(data: put(mea_score,2.), overrides: "&datafmt.");
obj.format_cell(data: put(gpvbp,4.1), overrides: "&datafmt.");
obj.format_cell(data: put(gpv90,3.), overrides: "&datafmt.");
obj.row_end();
```

In the DATA step observation loop, a row is started for each observation and displays the data in table

```
if eof then do;
        obj.table_end();
        obj.delete();
```

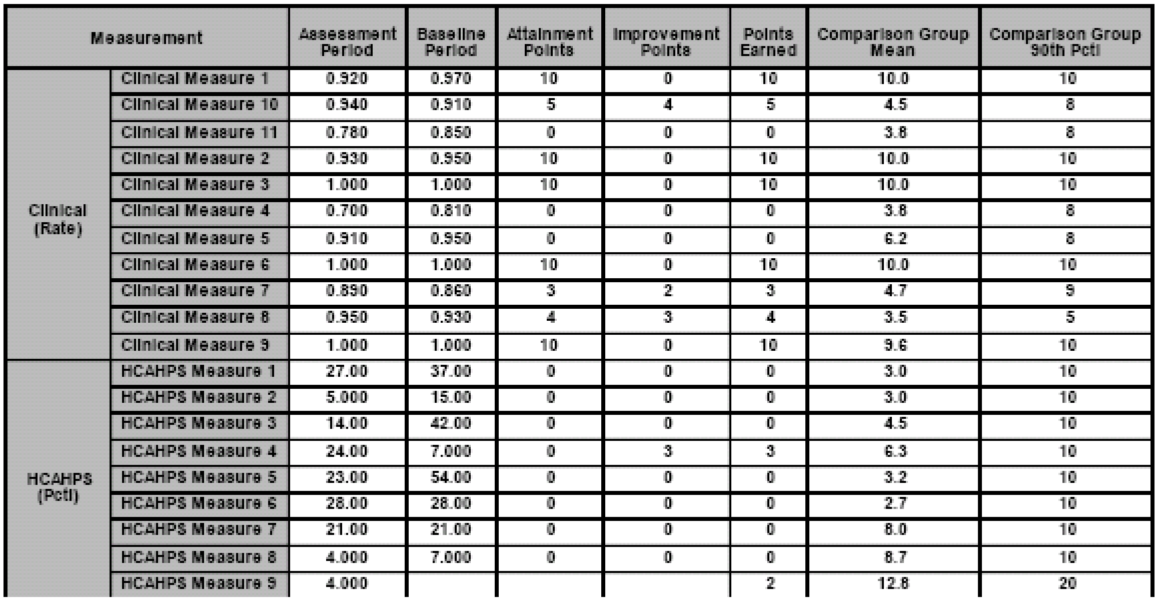
```
end;
run;
ods pdf close;
```

If eof=1, the observation loop is cycled through for the last time, tasks in termination execution phase are: end a table that was created during the initialization phase, and delete the ODS report writing class object. To detect if execution reaches end of file, use end= DATA step option. After all observations have been processed, close ODS outside of the DATA step.

EXAMPLE 2: ADD BACKGROUND COLOR TO ROW/COLUMN NAME, MERGE ROW/COLUMN CELLS

The table generated in example 1 contains all of the necessary data but is ineffective in communicating the information contained in the data.

In example 2, background color was added to the column name and row name, along with another column to show the two groups of measures: clinical measures and HCAHPS measures.



Measurement		Assessment Period	Baseline Period	Attainment Points	Improvement Points	Points Earned	Comparison Group Mean	Comparison Group 90th Pctl
Clinical (Rate)	Clinical Measure 1	0.920	0.970	10	0	10	10.0	10
	Clinical Measure 10	0.940	0.910	5	4	5	4.5	8
	Clinical Measure 11	0.780	0.850	0	0	0	3.8	8
	Clinical Measure 2	0.930	0.950	10	0	10	10.0	10
	Clinical Measure 3	1.000	1.000	10	0	10	10.0	10
	Clinical Measure 4	0.700	0.810	0	0	0	3.8	8
	Clinical Measure 5	0.910	0.950	0	0	0	6.2	8
	Clinical Measure 6	1.000	1.000	10	0	10	10.0	10
	Clinical Measure 7	0.890	0.860	3	2	3	4.7	9
	Clinical Measure 8	0.950	0.930	4	3	4	3.5	5
	Clinical Measure 9	1.000	1.000	10	0	10	9.6	10
HCAHPS (Pctl)	HCAHPS Measure 1	27.00	37.00	0	0	0	3.0	10
	HCAHPS Measure 2	5.000	15.00	0	0	0	3.0	10
	HCAHPS Measure 3	14.00	42.00	0	0	0	4.5	10
	HCAHPS Measure 4	24.00	7.000	0	3	3	6.3	10
	HCAHPS Measure 5	23.00	54.00	0	0	0	3.2	10
	HCAHPS Measure 6	28.00	28.00	0	0	0	2.7	10
	HCAHPS Measure 7	21.00	21.00	0	0	0	8.0	10
	HCAHPS Measure 8	4.000	7.000	0	0	0	8.7	10
	HCAHPS Measure 9	4.000				2	12.8	20

The code to generate the table above:

```
proc sql;
create table sumtbl_data as
select *, count(*) as mea_cnt
from sumtbl_data
group by measure_type;
quit;
```

This proc sql statement will add the clinical measure count and HCAHPS measure count to the data set; the counts will be used to determine how many rows should be merged into one row for the “Clinical (Rate)” and “HCAHPS (Pctl)”. The count may also be saved into macro variables clin_meacnt and hcahps_meacnt and be referred to later.

```
proc sql noprint;
select count(*) into :clin_meacnt from sumtbl_data where measure_type=’Clinical’; quit;
proc sql noprint;
select count(*) into :hcahps  meacnt from sumtbl  data where measure  type=’HCAHPS’; quit;
```

```
%let colfmt=background=greybb font_face=arial font_style=roman font_size=0.2 cellpadding=0 cellspacing=0
font_weight=bold just=center;

%let datafmt=font_face=arial font_style=roman font_size=0.2 cellheight=0.15in cellpadding=0 cellspacing=0
just=center font_weight=bold;
```

> Macro variable colfmt controls the format of column names; “background=greybb” was added to this macro variable to change the background color for column names.

```
options nodate nonumber center missing=' ' orientation=landscape;

ods noptitle;

ods noproctitle;

ods pdf file="ex2.pdf" notoc;

title; footnote;

proc sort data=sumtbl_data; by measure_type measure_name; run;


data _null_;

set sumtbl_data end=eof;

by measure_type measure_name;

if _N_ =1 then do;

        dcl odsout obj();

        obj.table_start(name: "VBP Report");

        obj.row_start();

        obj.format_cell(data: "Measurement", column_span:2, overrides: "&colfmt. cellheight=0.3in vjust=middle");
```

> A new column for “Clinical (Rate)” and “HCAHPS (Pctl)” will be added, the measurement column then need to span 2 columns. “just=center vjust=middle” option was set to display the text in the center of the box. Note that “just=center” has already set in macro &colfmt.

```
obj.format_cell(data: "Assessment*Period", split: '*', overrides: "&colfmt. cellheight=0.3in");

obj.format_cell(data: "Baseline*Period", split: '*',overrides: "&colfmt. cellheight=0.3in");

obj.format_cell(data: "Attainment*Points", split: '*',overrides: "&colfmt. cellheight=0.3in");

obj.format_cell(data: "Improvement*Points", split: '*', overrides: "&colfmt. cellheight=0.3in");

obj.format_cell(data: "Points*Earned", split: '*', overrides: "&colfmt. cellheight=0.3in");

obj.format_cell(data: "Comparison Group*Mean", split: '*', overrides: "&colfmt. cellheight=0.3in");

obj.format_cell(data: "Comparison Group *90th Pctl", split: '*', overrides: "&colfmt. cellheight=0.3in");

        obj.row_end();

end;


obj.row_start();

if first.measure_type & measure_type='Clinical' then do;
```

```
            obj.format_cell(data: "Clinical*(Rate)", split: '*', row_span:mea_cnt, overrides:
"backgroundcolor=greybb cellwidth=0.6in &datafmt. vjust=middle");
end;
if first.measure_type & measure_type='HCAHPS' then do;
            obj.format_cell(data: " HCAHPS*(Pctl)", split: '*', row_span:mea_cnt, overrides:
"backgroundcolor=greybb cellwidth=0.6in &datafmt. vjust=middle");
end;
```

> A new column for “Clinical (Rate)” and “HCAHPS (Pctl)” is added only once, it is done by checking the condition if first.measure_type. Note that mea_cnt (measure count) is used to control how many rows the cell will span.

```
obj.format_cell(data: trim(left(measure_name)), overrides: "background=greybb &datafmt. just=left
font_weight=bold");
```

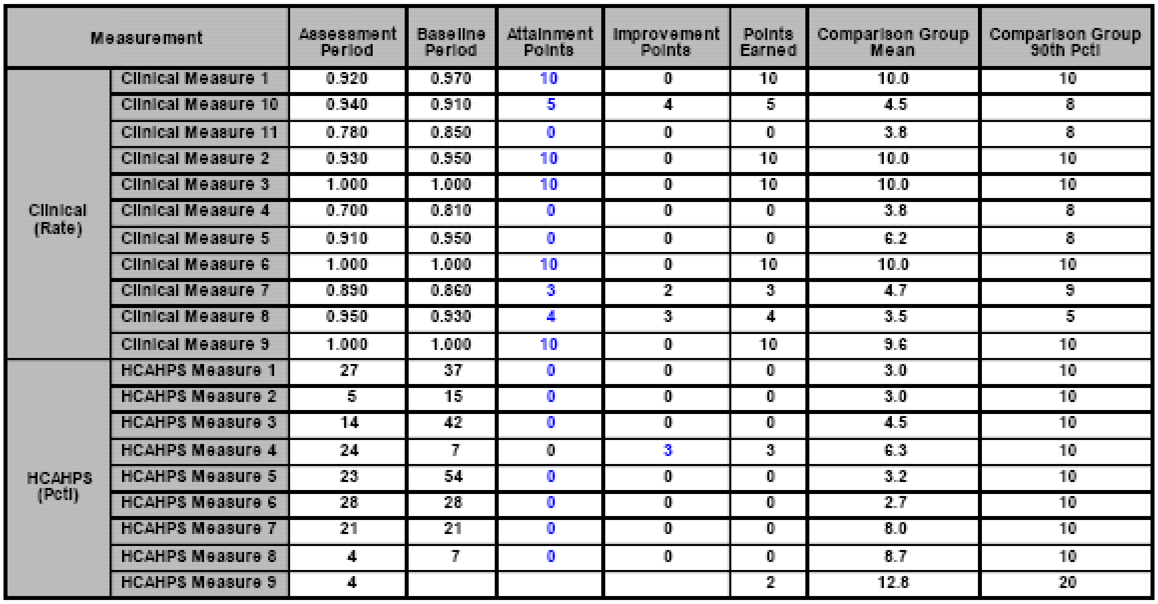
> Background color for measure name was added.

```
obj.format_cell(data: put(curr_rate, 5.3), overrides: "&datafmt.");
obj.format_cell(data: put(base_rate, 5.3), overrides: "&datafmt.");
obj.format_cell(data: put(att_score, 2.), overrides: "&datafmt.");
obj.format_cell(data: put(imp_score, 2.), overrides: "&datafmt.");
obj.format_cell(data: put(mea_score,2.), overrides: "&datafmt.");
obj.format_cell(data: put(gpvbp,4.1), overrides: "&datafmt.");
obj.format_cell(data: put(gpv90,3.), overrides: "&datafmt.");
obj.row_end();

if eof then do;
            obj.table_end();
            obj.delete();
end;
run;

ods pdf close;
```

EXAMPLE 3: CHANGE DATA FORMAT BASED ON ANOTHER COLUMN’S VALUE, ADD “TRAFFIC LIGHT”

The values in the assessment period and baseline period columns are scaled from 0 to 1 for clinical measures, and scaled from 0 to 100 for HCAHPS measures, because clinical measures values are proportional data, while HCAHPS measure values are percentile ranks. Decimal points for HCAHPS values in the assessment period and baseline period columns are not desired. The values should be displayed as #.### for clinical measures, and ### for HCAHPS measures as shown in the table below.

“Points Earned” is the maximum value of attainment points and improvement points and it would easier for users to visualize this information if the fields that the points are based on are highlighted. Attainment point values were highlighted in blue when assessment points were higher or equal to improvement points, and vice versa. The table below displays that feature:



Measurement		Assessment Period	Baseline Period	Attainment Points	Improvement Points	Points Earned	Comparison Group Mean	Comparison Group 90th Pctl
Clinical (Rate)	Clinical Measure 1	0.920	0.970	10	0	10	10.0	10
	Clinical Measure 10	0.940	0.910	5	4	5	4.5	8
	Clinical Measure 11	0.780	0.850	0	0	0	3.8	8
	Clinical Measure 2	0.930	0.950	10	0	10	10.0	10
	Clinical Measure 3	1.000	1.000	10	0	10	10.0	10
	Clinical Measure 4	0.700	0.810	0	0	0	3.8	8
	Clinical Measure 5	0.910	0.950	0	0	0	6.2	8
	Clinical Measure 6	1.000	1.000	10	0	10	10.0	10
	Clinical Measure 7	0.890	0.860	3	2	3	4.7	9
	Clinical Measure 8	0.950	0.930	4	3	4	3.5	5
	Clinical Measure 9	1.000	1.000	10	0	10	9.6	10
HCAHPS (Pctl)	HCAHPS Measure 1	27	37	0	0	0	3.0	10
	HCAHPS Measure 2	5	15	0	0	0	3.0	10
	HCAHPS Measure 3	14	42	0	0	0	4.5	10
	HCAHPS Measure 4	24	7	0	3	3	6.3	10
	HCAHPS Measure 5	23	54	0	0	0	3.2	10
	HCAHPS Measure 6	28	28	0	0	0	2.7	10
	HCAHPS Measure 7	21	21	0	0	0	8.0	10
	HCAHPS Measure 8	4	7	0	0	0	8.7	10
	HCAHPS Measure 9	4				2	12.8	20

The code to generate the table above:

```
%let colfmt=background=greybb font_face=arial font_style=roman font_size=0.2 cellpadding=0 cellspacing=0 font_weight=bold just=center;

%let datafmt=font_face=arial font_style=roman font_size=0.2 cellheight=0.15in cellpadding=0 cellspacing=0 just=center font_weight=bold;


options nodate nonumber center missing=' ' orientation=landscape;

ods noptitle;

ods noproctitle;

ods pdf file="ex3.pdf" notoc;

title; footnote;


proc sort data=sumtbl_data; by measure_type measure_name; run;


data _null_;

set sumtbl_data end=eof;

by measure_type measure_name;

if _N_ =1 then do;

        dcl odsout obj();

        obj.table_start(name: "VBP Report");

        obj.row_start();

        obj.format_cell(data: "Measurement", column_span:2, overrides: "&colfmt. cellheight=0.3in vjust=middle");

        obj.format_cell(data: "Assessment*Period", split: '*', overrides: "&colfmt. cellheight=0.3in");

        obj.format_cell(data: "Baseline*Period", split: '*',overrides: "&colfmt. cellheight=0.3in");

        obj.format_cell(data: "Attainment*Points", split: '*',overrides: "&colfmt. cellheight=0.3in");
        obj.format_cell(data: "Improvement*Points", split: '*', overrides: "&colfmt. cellheight=0.3in");
        obj.format_cell(data: "Points*Earned", split: '*', overrides: "&colfmt. cellheight=0.3in");
        obj.format_cell(data: "Comparison Group*Mean", split: '*', overrides: "&colfmt. cellheight=0.3in");
        obj.format_cell(data: "Comparison Group *90th Pctl", split: '*', overrides: "&colfmt. cellheight=0.3in");
        obj.row_end();
end;

obj.row_start();
if first.measure_type & measure_type='Clinical' then do;
        obj.format_cell(data: "Clinical*(Rate)", split: '*', row_span:mea_cnt, overrides: "backgroundcolor=greybb
cellwidth=0.6in &datafmt. vjust=middle");
end;
if first.measure_type & measure_type='HCAHPS' then do;
        obj.format_cell(data: " HCAHPS*(Pctl)", split: '*', row_span:mea_cnt, overrides: "backgroundcolor=greybb
cellwidth=0.6in &datafmt. vjust=middle");
end;
obj.format_cell(data: trim(left(measure_name)), overrides: "background=greybb &datafmt. just=left
font_weight=bold");
if measure_type='Clinical' then do;
        obj.format_cell(data: compress(put(curr_rate, 5.3)), overrides: "&datafmt.");
        obj.format_cell(data: compress(put(base_rate, 5.3)), overrides: "&datafmt.");
end;
if measure_type='HCAHPS' then do;
        obj.format_cell(data: compress(put(curr_rate, 3.)), overrides: "&datafmt.");
        obj.format_cell(data: compress(put(base_rate, 3.)), overrides: "&datafmt.");
end;
```

If the observation is for clinical measures, the decimal points are shown: put(curr_rate, 5.3),:otherwise, show as integer: put(curr_rate, 3.)

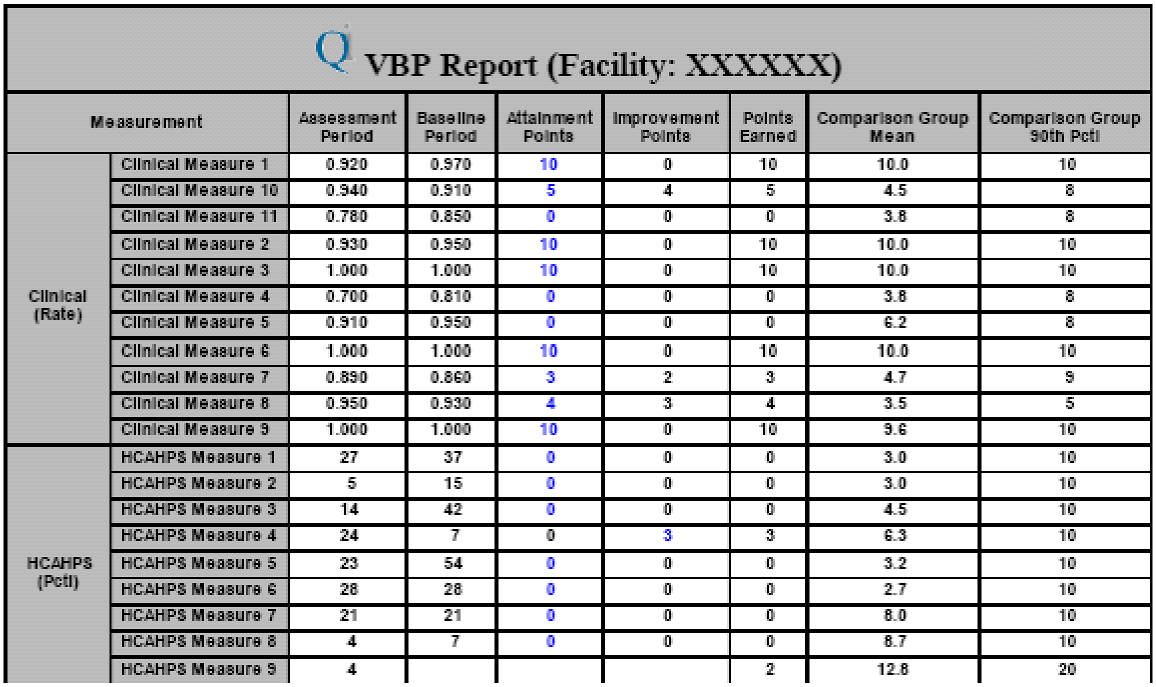
```
if att_score ge imp_score then do;
        obj.format_cell(data: put(att_score, 2.), overrides: "&datafmt. color=blue");
        obj.format_cell(data: put(imp_score, 2.), overrides: "&datafmt.");
end;
else do;
        obj.format_cell(data: put(att_score, 2.), overrides: "&datafmt.");
        obj.format_cell(data: put(imp_score, 2.), overrides: "&datafmt. color=blue");
end;
```

Highlight the attainment point value in blue when it is higher or equal to improvement points and vice versa.

```
obj.format_cell(data: put(mea_score,2.), overrides: "&datafmt.");
obj.format_cell(data: put(gpvbp,4.1), overrides: "&datafmt.");
obj.format_cell(data: put(gpv90,3.), overrides: "&datafmt.");
obj.row_end();

if eof then do;
        obj.table_end();
        obj.delete();
end;
run;

ods pdf close;
```

EXAMPLE 4: ADD EMBEDDED IMAGES

A logo of the organization was added to the table header row. The preimage option embeds the image in front of the cell text, postimage will embed the image after the text. The following table shows an example of using preimage.



VBP Report (Facility: XXXXXX)								
Measurement		Assessment Period	Baseline Period	Attainment Points	Improvement Points	Points Earned	Comparison Group Mean	Comparison Group 90th Pctl
Clinical (Rate)	Clinical Measure 1	0.920	0.970	10	0	10	10.0	10
	Clinical Measure 10	0.940	0.910	5	4	5	4.5	8
	Clinical Measure 11	0.780	0.850	0	0	0	3.8	8
	Clinical Measure 2	0.930	0.950	10	0	10	10.0	10
	Clinical Measure 3	1.000	1.000	10	0	10	10.0	10
	Clinical Measure 4	0.700	0.810	0	0	0	3.8	8
	Clinical Measure 5	0.910	0.950	0	0	0	6.2	8
	Clinical Measure 6	1.000	1.000	10	0	10	10.0	10
	Clinical Measure 7	0.890	0.860	3	2	3	4.7	9
	Clinical Measure 8	0.950	0.930	4	3	4	3.5	5
	Clinical Measure 9	1.000	1.000	10	0	10	9.6	10
HCAHPS (Pctl)	HCAHPS Measure 1	27	37	0	0	0	3.0	10
	HCAHPS Measure 2	5	15	0	0	0	3.0	10
	HCAHPS Measure 3	14	42	0	0	0	4.5	10
	HCAHPS Measure 4	24	7	0	3	3	6.3	10
	HCAHPS Measure 5	23	54	0	0	0	3.2	10
	HCAHPS Measure 6	28	28	0	0	0	2.7	10
	HCAHPS Measure 7	21	21	0	0	0	8.0	10
	HCAHPS Measure 8	4	7	0	0	0	8.7	10
	HCAHPS Measure 9	4				2	12.8	20

The code to generate the table above :

```
%let colfmt=background=greybb font_face=arial font_style=roman font_size=0.2 cellpadding=0 cellspacing=0 font_weight=bold just=center;

%let datafmt=font_face=arial font_style=roman font_size=0.2 cellheight=0.15in cellpadding=0 cellspacing=0 just=center font_weight=bold;


options nodate nonumber center missing=' ' orientation=landscape;
ods noptitle;
ods noproctitle;
ods pdf file="ex4.pdf" notoc;
title; footnote;

proc sort data=sumtbl_data; by measure_type measure_name; run;

data _null_;
set sumtbl_data end=eof;
by measure_type measure_name;
if _N_ =1 then do;
        dcl odsout obj();
        obj.table_start(name: "VBP Report");
        obj.row_start();
        obj.format_cell(data: "VBP Report (Facility: " || trim(left(hospital_name)) || ")", column_span:9, overrides: "preimage='C:\images\logo.png' backgroundcolor=greybb font_size=5 font_weight=bold just=center");
        obj.row_end();
```

Use preimage= option to add logo image in front of the header row.

```
        obj.row_start();
        obj.format_cell(data: "Measurement", column_span:2, overrides: "&colfmt. cellheight=0.3in vjust=middle");
        obj.format_cell(data: "Assessment*Period", split: '*', overrides: "&colfmt. cellheight=0.3in");
        obj.format_cell(data: "Baseline*Period", split: '*',overrides: "&colfmt. cellheight=0.3in");
        obj.format_cell(data: "Attainment*Points", split: '*',overrides: "&colfmt. cellheight=0.3in");
        obj.format_cell(data: "Improvement*Points", split: '*', overrides: "&colfmt. cellheight=0.3in");
        obj.format_cell(data: "Points*Earned", split: '*', overrides: "&colfmt. cellheight=0.3in");
        obj.format_cell(data: "Comparison Group*Mean", split: '*', overrides: "&colfmt. cellheight=0.3in");
        obj.format_cell(data: "Comparison Group *90th Pctl", split: '*', overrides: "&colfmt. cellheight=0.3in");
        obj.row_end();
        obj.head_end();
end;

obj.row_start();
if first.measure_type & measure_type='Clinical' then do;
        obj.format_cell(data: "Clinical*(Rate)", split: '*', row_span:mea_cnt, overrides: "backgroundcolor=greybb cellwidth=0.6in &datafmt. vjust=middle");
end;
if first.measure_type & measure_type='HCAHPS' then do;
        obj.format_cell(data: " HCAHPS*(Pctl)", split: '*', row_span:mea_cnt, overrides: "backgroundcolor=greybb cellwidth=0.6in &datafmt. vjust=middle");
end;
```

```
obj.format_cell(data: trim(left(measure_name)), overrides: "background=greybb &datafmt. just=left
font_weight=bold");
if measure_type='Clinical' then do;
        obj.format_cell(data: put(curr_rate, 5.3), overrides: "&datafmt.");
        obj.format_cell(data: put(base_rate, 5.3), overrides: "&datafmt.");
end;
if measure_type='HCAHPS' then do;
        obj.format_cell(data: put(curr_rate, 3.), overrides: "&datafmt.");
        obj.format_cell(data: put(base_rate, 3.), overrides: "&datafmt.");
end;
if att_score ge imp_score then do;
        obj.format_cell(data: put(att_score, 2.), overrides: "&datafmt. color=blue");
        obj.format_cell(data: put(imp_score, 2.), overrides: "&datafmt.");
end;
else do;
        obj.format_cell(data: put(att_score, 2.), overrides: "&datafmt.");
        obj.format_cell(data: put(imp_score, 2.), overrides: "&datafmt. color=blue");
end;
obj.format_cell(data: put(mea_score,2.), overrides: "&datafmt.");
obj.format_cell(data: put(gpvbp,4.1), overrides: "&datafmt.");
obj.format_cell(data: put(gpv90,3.), overrides: "&datafmt.");
obj.row_end();

if eof then do;
        obj.table_end();
        obj.delete();
end;
run;

ods pdf close;
```

CONCLUSION

Data _NULL_ report writing using ODS OO features is a very powerful and easy to use tool to create custom tables. Combining the power of DATA step functionalities and ODS report writing object functionalities, this paper demonstrates how easily a plain table can be turned into an attractive and more effective table to communicate complex data. The only caveat is the warning message: ”WARNING: DATA step interface is preproduction in this release”, because this feature is still in preproduction at the time this paper was prepared.

REFERENCES

Daniel O’Connor, SAS Institute Inc. - The Power to Show: Ad Hoc Reporting, Custom Invoices, and Form Letters

Daniel O’Connor, SAS Institute Inc., Cary NC - Next Generation Data _NULL_ Report Writing Using ODS OO Features

SAS Online Documentation: http://support.sas.com/rnd/base/datastep/dsobject/index.html

ACKNOWLEDGMENTS

Nikolas Matthes, MD, PhD, MPH, Vice President for Research at the Quality Indicator Project®, provided overall leadership for the VBP reporting project.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Alice Liqiong Li, MS
SAS Programmer
Research and Development
Quality Indicator Project, Maryland Hospital Association
6820 Deerpath Road
Elkridge, MD 21075
Email: ali@mhaonline.org

Jacob Jen-Hao Cheng, MS, PhD
Research Manager and Statistician
Research and Development
Quality Indicator Project, Maryland Hospital Association
6820 Deerpath Road
Elkridge, MD 21075
Email: jcheng@mhaonline.org