

Using SAS Macro Language and PROC Report to produce formatted Excel files

Nadeene Blanchard-Martin
SAS Certified Base Programmer for SAS 9
Master of Science, Statistics
Data Analyst (on mat leave), MOHLTC

Problem

- Hospitals uploaded infection statistics to a web site with a backend database that has no field/column validation
- As a result, some fields have both numeric and character data within the same field
- Data from all hospitals are downloaded into 1 Excel file and need to be cleaned then compiled into 1 Excel report for disbursal

Original Input Excel File

- The excel file with the combined data from all the hospitals was laid out as follows:
 - Rows 1 & 2 had text.
 - Row 3 had numbers in sequential order starting from 1, corresponding to the column number.

Original Input Excel File Continued

- Column names started on row 4.
- Data started on row 5.
- Each hospital could have up to 6 sites so the data was laid out in wide format (as opposed to long format).
 - There was 1 row per hospital and 6 columns for each field to accommodate hospitals with multiple sites.

Original Input Excel File

Continued

- Columns included hospital id, name, type, region, 6 report date columns, 6 person reporting columns, 6 email contact columns, 6 reporting period columns, 6 site number columns, 6 site name columns etc.
- The statistical data for each hospital started in column BB - there were 6 columns for each statistic.

Stage 1 SAS Macro

- A 'stage 1' SAS macro read in the Excel file with data from all the hospitals, cleaned it and output a new excel file with various flags and calculation checks.

Picture of the Excel XML file produced by Stage 1 Macro

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Facility ID	Institution ID	Email Contact		New Cases	New Cases-PROBLEM		New Cases Other	New Cases Other-PROBLEM		New Cases Unknown	New Cases Unknown-PROBLEM		# of Patient Days Reported	Mean of Patient Days of Last 3 Months	Percentage Difference
2	100	4600														

	A	B	C	D	E	F	G	H	I
1	Facility ID	Institution ID	Email Contact		New Cases	New Cases-PROBLEM		New Cases Other	New Cases Other-PROBLEM
2	100	4600	info@hospital0.com		.			.	
3	101	4356	info@hospital1.com		1			0	

108	1717	info@hospital.com		1		1		1		8072	7417.33	8.83%
-----	------	--	--	---	--	---	--	---	--	------	---------	-------

	K	L	M	N	O	P	Q
1	New Cases Unknown	New Cases Unknown-PROBLEM		# of Patient Days Reported	Mean of Patient Days of Last 3 Months	Percentage Difference	Patient Days-PROBLEM
2	
3				1015	1100.00	1.0000	

Excel XML file produced by Stage 1 Macro

- Certain cells were colour coded based on specific criteria
- Columns A and B were frozen
- Black background columns inserted for separation purposes
- 'Problem' columns added for data checking

How the stage 1 Excel XML file is used

- Analyst verified data with hospitals.
- Once data was verified and the stage 1 Excel file was updated, analyst ran a 'stage 2' SAS macro with this updated Excel file as input.

Stage 2 SAS Macro

- This stage 2 SAS macro formats the updated stage 1 excel file into the final Excel file for public disbursement.
- This presentation will illustrate the code in the stage 2 SAS macro.

Stage 2 SAS Macro

- User runs stage 2 SAS macro.
- Stage 2 macro first generates a pop up window for user to enter parameters about the Excel file.

Code within Stage 2 macro
- customizes the menu

```
libname proclib 'C:\New SAS Macros\';  
proc pmenu catalog=proclib.menus;  
menu select;  
  item 'File' menu=f;  
  item 'Edit' submenu=editmnu;  
    menu f;  
    item 'End this window' selection=endwdw;  
    item 'End this SAS session'  
      selection=endsas;  
      selection endwdw 'end';  
      selection endsas 'bye';  
    submenu editmnu sashelp.core.edit;  
run; quit;
```

Code within stage 2 macro

-creates the popup window to get data

```
%window userInput irow=5 rows=25 icolumn=15  
columns=55 menu=proclib.menus.select
```

```
#1 @5 "CONVERT CDI CHECKING FILE TO FINAL FILE"  
attr = highlight
```

```
#3 @5 "-----"
```

```
#4 @5 "Press ENTER to begin, fill in the info then press  
ENTER after each line. "
```

```
#6 @5 "Indicator (CDI only): " +2 indicator 100 attr =  
underline required=yes
```

Code within stage 2 macro
-creates the popup window to get data

**#8 @5 "Path=Directory where checking file is located.
Ensure it ends in \."**

#9 @5 "Example: C:\PT SAFETY\"

**#10 @5 "Path of Excel File:" +2 Path 105 attr =
underline required=yes**

**#12 @5 "File=filename of the xls file - must end in xls
and be an excel file I.E. have xls extension "**

Code within stage 2 macro

-creates the popup window to get data

**#13 @5 "File Name of Excel File:" +2 File 100 attr =
underline required=yes**

**#15 @5 "Month (Example 1,2,3,...,12):" +2 Month 100
attr = underline required=yes**

**#17 @5 "Year (between 2008 and 2108):" +2 Year 100
attr = underline required=yes;**

Code within stage 2 macro
-display the popup window to get data

```
/*get user input*/
```

```
%display UserInput;
```

Result of this code

CONVERT CDI CHECKING FILE TO FINAL FILE

Press ENTER to begin, fill in the info then press ENTER after each line.

Indicator (CDI only): _____

Path=Directory where checking file is located. Ensure it ends in \.
Example: C:\Documents and Settings\blanchardna\Desktop\PT SAFETY\
Path of Excel File: _____

File=filename of the xls file - must end in xls and be an excel file I.E. have xls extension
File Name of Excel File: _____

Month (Example 1,2,3,...,12): _____

Year (between 2008 and 2108): _____

User Enters Parameter Values

SAS - [USERINPUT]

File Edit Window

CONVERT CDI CHECKING FILE TO FINAL FILE

Press ENTER to begin, fill in the info then press ENTER after each line.

Indicator (CDI only): CDI

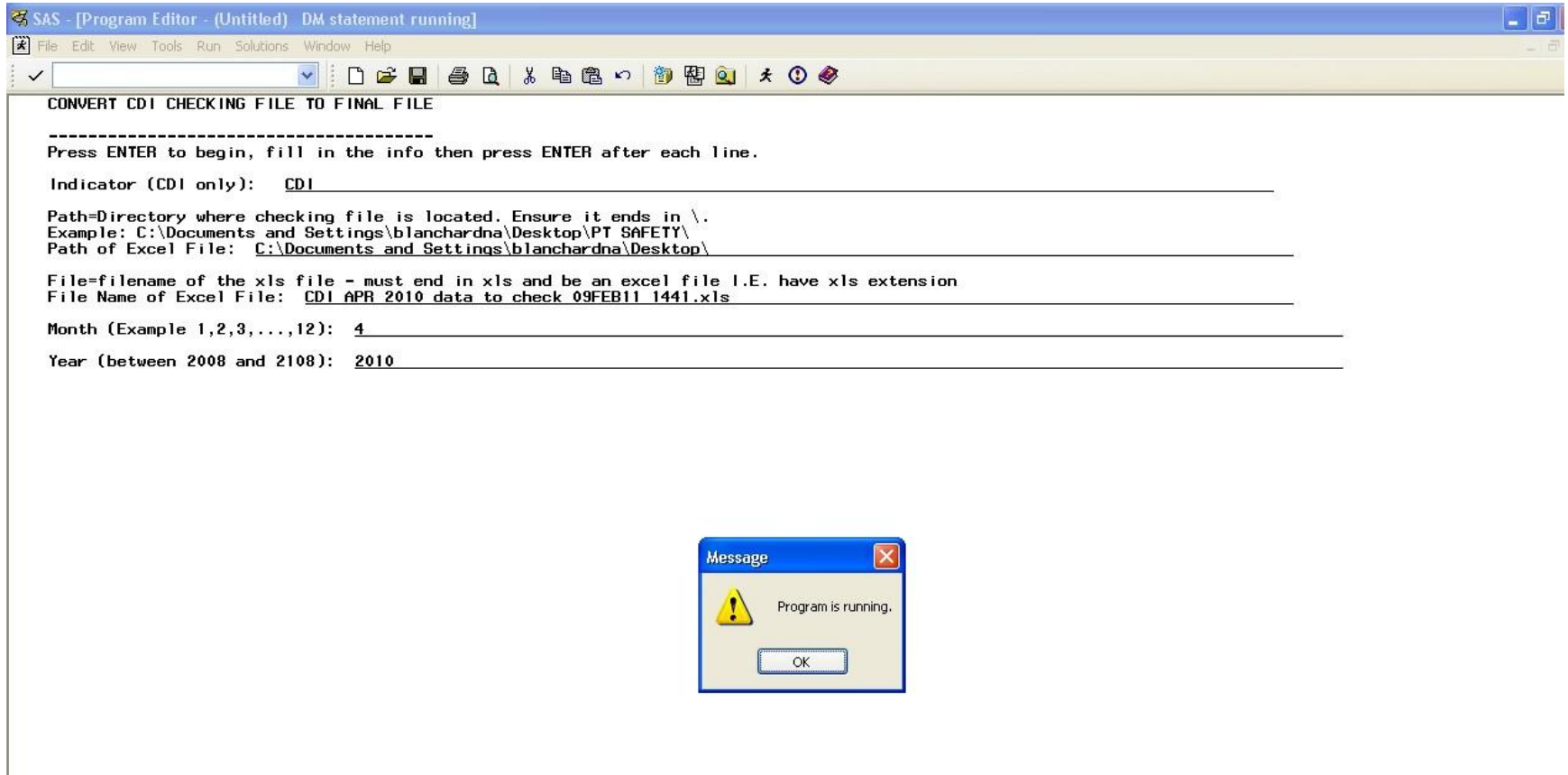
Path=Directory where checking file is located. Ensure it ends in \.
Example: C:\Documents and Settings\blanchardna\Desktop\PT SAFETY\
Path of Excel File: C:\Documents and Settings\blanchardna\Desktop\

File=filename of the xls file - must end in xls and be an excel file I.E. have xls extension
File Name of Excel File: CDI APR 2010 data to check 09FEB11 1441.xls

Month (Example 1,2,3,...,12): 4

Year (between 2008 and 2108): 2010

A pop up box appears to alert user that program is running



Code to generate pop up box

```
dm "POSTMESSAGE 'Program is running.'";  
data;  
run;  
quit;
```

Code that prints values entered in log

```
/*
```

To confirm the user-defined macro variables:

```
*/
```

```
%put NOTE: You defined the following macro  
variables: ;
```

```
%put NOTE: Indicator=&indicator;
```

```
%put NOTE: Path=&Path;
```

```
%put NOTE: File=&File;
```

```
%put NOTE: Month=&Month;
```

```
%put NOTE: Year=&Year;
```

Error checking for blank macro parameters

Example:

```
/*To ensure all macro parameters were specified:*/  
%if %length(&indicator) = 0 %then %do ;  
    %put ERROR: Value for macro parameter indicator  
    is missing ;  
    dm "POSTMESSAGE 'Value for macro parameter  
    indicator is missing.'";  
    data; run;quit;  
    %goto finish ;  
%end ;
```

This is done for all parameters.

Code to ensure the path specified ends in backslash

```
%LET length_of_path= %sysevalf(%length(%trim(&path)),integer);  
%put NOTE: length of path = &length_of_path;  
%put NOTE: last character of path =  
    %substr(%trim(&path),&length_of_path);  
%if %substr(%trim(&path),&length_of_path) ne \ %then %do;  
    %put *****;  
    %put ERROR: THE PATH MUST END IN BACK SLASH \.;  
    dm "POSTMESSAGE 'THE PATH MUST END IN BACK SLASH \.'";  
    data; run; quit;  
    %put *****;  
    %goto finish ;  
%end;
```

Code to remove white space from around parameters

```
%let indicator=%trim(&indicator);
```

```
%let Path=%trim(&path);
```

```
%let File=%trim(&file);
```

```
%let Month=%trim(&Month);
```

```
%let Year=%trim(&year);
```

Code to ensure month is numeric

```
%if not(%datatyp(&month)=NUMERIC) %then %do;  
  %put ERROR: *****,  
  %put ERROR: Month must be numeric - i.e. one of  
  these: 1,2,3,4,5,6,7,8,9,10,11,12;  
  dm "POSTMESSAGE 'Month must be numeric i.e. a  
  number from 1 to 12.'";  
  data;  
  run; quit;  
  %put ERROR: *****,  
  %goto finish ;  
%end ;
```

Code to ensure month is between 1 and 12

```
%if not(&month in 1 2 3 4 5 6 7 8 9 10 11 12) %then %do;  
%put *****;  
%put ERROR: Month must be numeric - i.e. one of  
these: 1,2,3,4,5,6,7,8,9,10,11,12;  
dm "POSTMESSAGE 'Month must be numeric i.e. a  
number from 1 to 12.'";  
data;  
run; quit;  
%put *****;  
%goto finish ;  
%end ;
```

Code to assign names to month

```
%if &month. GE 1 and %month. LE 12 %then %do;  
  %LET monthname=%sysfunc(upcase(%sysfunc(putn(  
    %sysfunc(mdy(&month.,1,2011)),worddate3.));  
%end;  
%else %do;  
  %put *****;  
  %put ERROR: Month must be numeric - i.e. one of  
these: 1,2,3,4,5,6,7,8,9,10,11,12;  
  dm "POSTMESSAGE 'Month must be numeric i.e. a  
  number from 1 to 12.'";  
  data;  
  run;  
  quit;  
  %put *****;  
  %goto finish ;  
%end;
```

Code to ensure that correct value for indicator parameter was entered

```
%if %upcase(&indicator) ne CDI %then %do ;  
  %put ERROR: This is only for CDI NOT &indicator ;  
  dm "POSTMESSAGE 'This macro is only for CDI.'";  
  data; run; quit;  
  %goto finish ;  
%end ;  
%let indicator=&indicator;
```

Code to ensure that correct value for indicator parameter was entered

```
%if %upcase(&indicator) ne CDI %then %do ;  
  %put ERROR: This is only for CDI NOT &indicator ;  
  dm "POSTMESSAGE 'This macro is only for CDI.'";  
  data; run; quit;  
  %goto finish ;  
%end ;  
%let indicator=&indicator;  
%let extension=%substr(&File,%eval(%length(&File))-2) ;  
%put NOTE: &extension ;  
%put NOTE: Length of extension: %length(&extension);  
%if %upcase(&extension) ne XLS %then %do;  
  %put ERROR: The File supplied is not an XLS file. ;  
  dm "POSTMESSAGE 'The File supplied is not an XLS file.'";  
  data; run;quit;  
  %goto finish ;  
%end ;
```

Code to ensure year entered was numeric

```
%if not(%datatyp(&year)=NUMERIC) %then %do;  
  %put ERROR: *****,  
  %put ERROR: Value for macro parameter Year should  
  be between 2008 and 2108 ;  
  dm "POSTMESSAGE 'Value for macro parameter Year  
  should be between 2008 and 2108.'";  
  data; run; quit;  
  %put ERROR: *****,  
  %goto finish ;  
%end ;
```

Code to ensure year was between 2008 and 2108

```
%if &Year lt 2008 or &Year gt 2108 %then  
%do ;  
    %put ERROR: Value for macro parameter Year  
    should be between 2008 and 2108 ;  
    dm "POSTMESSAGE 'Value for macro parameter  
    Year should be between 2008 and 2108.'";  
    data; run; quit;  
    %goto finish ;  
%end ;
```

After all these checks:

Excel file is imported with proc import.

```
PROC IMPORT OUT= WORK.COMPARISON2
```

```
DATAFILE= "&Path.&File"
```

```
DBMS=EXCEL REPLACE;
```

```
GETNAMES=YES;
```

```
MIXED=NO;
```

```
SCANTEXT=YES;
```

```
USEDATE=YES;
```

```
SCANTIME=YES;
```

```
RUN;
```

```
QUIT;
```

Code to ensure data set got created from
imported excel file

```
%if %sysfunc(exist(COMPARISON2)) = 0 %then
```

```
%do ;
```

```
  %put ERROR: data set comparison2, created  
  from the imported file, does not exist ;
```

```
  %goto finish ;
```

```
%end ;
```

```
%else %put NOTE: data set comparison2,  
  created from the imported file, exists;
```

Code to ensure data set created from imported excel file has >0 observations

```
%global num;  
%let num=0;  
proc sql noprint;  
  select (nobs-delobs) into :num  
  from dictionary.tables  
  where libname="%upcase(work)"  
  and memname = "%upcase(comparison2)";  
Quit;
```

Code to ensure data set created from imported excel file has >0 observations

```
%let num=&num;
```

```
%if &num=0 %then %do;
```

```
    %put NOTE: The data set %upcase(comparison2) in  
    library %upcase(work) is empty or does not exist.;
```

```
    %goto finish ;
```

```
    %end;
```

```
%else %put NOTE: The data set %upcase(comparison2)  
    in library %upcase(work) has &num observations.;
```

Data Cleaning

- Data is cleaned by deleting blank records.
- Master list of hospitals is imported and merged with data set.
- Case of certain text fields are adjusted using propcase() function.
- Using past months' data that is stored in a SAS data set, records which exceed the 80th percentile this month, based on last 3 months of data are flagged.
- Other flagging is done.
- Excel Report is created.

Creation of the Report

- Set paper size

```
/*set paper size to legal*/
```

```
options papersize=legal;
```

```
/*print out the paper size setting in log*/
```

```
proc options
```

```
  option=papersize;
```

```
run;
```

```
quit;
```

Set templates to be used by proc report & ODS Excel XP Tagset

```
libname mytpls '.';
```

```
ods path (prepend) mytpls.template(update);
```

```
libname mytpls clear;
```

Set templates to be used by proc report & ODS Excel XP Tagset

```
proc template;  
  define style styles.test;  
    parent=styles.minimal;  
    style test1 from rowheader / vjust=middle  
    JUST=CENTER;  
    style test2 from rowheader / vjust=middle  
    JUST=CENTER TAGATTR='format:0.00';  
    style test3 from rowheader / vjust=middle  
    JUST=CENTER TAGATTR='format:#,##0';  
  end;  
run; quit;
```

Set Formats to be used later

```
proc format;  
  value period  
  .= 'NOT REPORTED'  
  ;  
  value ratefmt  
  .= 'NOT REPORTED'  
  OTHER= [8.2]  
  ;  
run;  
quit;
```

Creation of the report

- Uses proc report and ODS Excel XP tagset to write the report out to a XML file readable by Excel

Set the title that will appear
in the Excel report

```
title "&indicator Reporting Period:  
&Monthname &Year";
```

Set the filename of the report

```
%let
```

```
fname1=&Path.&indicator._All_&monthname.  
_&Year._PHD.xml;
```

Include the new ODS Excel XP tagset

- This Tagset/Destination creates Microsoft's spreadsheet markup language XML.
- It is used specifically for importing data into Excel.
- Latest version was obtained from Chevell Parker of SAS Tech Support in Feb 2011 but SAS 9.1.3 onwards has a built in version.

Include the new ODS Excel XP tagset

- Code:

```
/*add the new tagset to the work library*/;
```

```
ods path(prepend) work.templat(update);
```

```
%include "C:\excltags_122.tpl";
```

To see the help and all options for Ods Excel XP tagset in the log

```
ods tagsets.excelxp file='test.xml'  
  options(doc="help");  
ods tagsets.excelxp CLOSE;
```

This will print out the help file for the tagset in the log.

ODS Excel XP tagset statement

```
ods tagsets.excelxp file="&fname1" style=Styles.test  
options(  

```

```
    Orientation='Landscape'
```

```
    skip_space='3,2,0,0,1'
```

```
    CENTER_VERTICAL='yes'
```

```
    CENTER_HORIZONTAL='yes'
```

```
    ABSOLUTE_Column_Width='6.76,8.3,23,24.75,9,8,7,  
    7,7,8,13.42,17.92,5.67,12.17'
```

```
    Autofit_height='yes'
```

ODS Excel XP tagset statement

Pages_FitWidth='1'

Pages_FitHeight='13'

FROZEN_HEADERS='yes'

FROZEN_ROWHEADERS='yes'

GRIDLINES='yes'

sheet_interval='none'

sheet_name="&indicator"

row_repeat='header'

Embedded_Footnotes='yes'

ODS Excel XP tagset statement

```
Print_Footer='&L&8Health Analytics  
Branch HSIMI &C &R &8Page  
&P of &N&#13;&8Last Updated:  
&D'
```

```
);
```

Proc report statement

```
proc report data=comparison3 nowindows split='*'
```

```
    style(column)=[BACKGROUND=white  
    BORDERCOLOR=black BORDERWIDTH=1  
    VERTICALALIGN=MIDDLE TEXTALIGN=CENTER  
    vjust=middle JUST=CENTER FONT=(Verdana, 10pt,  
    NORMAL)]
```

Proc report statement

```
style(header)=[BACKGROUND=light grey  
BORDER=black BORDERWIDTH=1  
VERTICALALIGN=MIDDLE TEXTALIGN=CENTER  
vjust=middle JUST=CENTER FONT=(Verdana, 10pt,  
BOLD)]
```

```
style(summary)=[BACKGROUND=white  
BORDER=black BORDERWIDTH=1  
VERTICALALIGN=MIDDLE TEXTALIGN=CENTER  
vjust=middle JUST=CENTER FONT=(Verdana, 10pt,  
BOLD)];
```

Proc Report – column statement

column

```
facility_id Institution_ID  
hospital_corporation hospital_site  
new_cases newcases  
new_cases_other newcasesother  
new_cases_unknown newcasesunknown  
__of_Patient_Days_Reported ptdaysfinal  
Rate_Calculated_by_SAS ratefinal  
percentile percentile=pctl  
outbreak Hospital_Group lhin lhin_code city;
```

Define statements for display variables

```
define facility_id / display order=data  
'Facility*Number' style=test1 ;  
  
define Institution_ID / display order=data  
'Institution*Number' style=test1 ;  
  
define hospital_corporation / display  
'Hospital*Corporation' style=test1 ;  
  
define hospital_site / display  
'Hospital*Site*Name' style=test1 ;
```

Define statement for column with computation

```
define new_cases / analysis noprint  
  'Cases*Reporting*Facility' style=test1 ;  
define newcases / computed  
  'Cases*Reporting*Facility' style=test1 ;
```

Compute statement for this column

```
compute newcases / char length=12;
```

```
  if _break_ne " "
```

```
  then call define( _col_,  
"style", "style={tagattr='formula:SUM(R2C:R[-1]C)'}");
```

*R[-1]C Reference the value in the
cell 1 row up and in the same column.;

*R2C reference the value in row 2 of
the same column.;

Compute statement for this column

```
    else if (facility_id=862 and institution_id=4632)
or (facility_id=111 and institution_id=4628)
    or (facility_id=958 and institution_id=4047) or
(facility_id=958 and institution_id=4049)
    or (facility_id=958 and institution_id=4080) or
(facility_id=632 and institution_id=4234)
    or (facility_id=962 and institution_id=4614)
    then newcases='N/A';
    else if LEFT( PUT(new_cases.sum , 8.0))=. then
newcases='NOT REPORTED';
    else newcases=LEFT( PUT(new_cases.sum , 8.0));
endcomp;
```

Define statement for column with computation

```
define Rate_Calculated_by_SAS/ analysis sum  
noprnt 'Calc. Rate' style=test2 ;  
define ratefinal / computed 'Calc. Rate'  
style=test2;
```

Compute block for this column

```
compute ratefinal / char length=12;  
if _break_ ne " " then do;  
  call define( _col_, "style", "style={tagattr='format:0.00  
formula:(RC[-4]/RC[-1])*1000'}");  
  call define( _col_,  
"style/merge", "style=[VERTICALALIGN=MIDDLE]");  
*vertical alignment = centre;
```

Compute block for this column

```
call define( _col_,  
  "style/merge", "style=[TEXTALIGN=CENTER]");  
  *horizontal alignment = centre;  
end;  
else if (facility_id=862 and institution_id=4632) or  
  (facility_id=111 and institution_id=4628)  
  or (facility_id=958 and institution_id=4047) or  
  (facility_id=958 and institution_id=4049)  
  or (facility_id=958 and institution_id=4080) or  
  (facility_id=632 and institution_id=4234)
```

Compute block for this column

```
    or (facility_id=962 and
institution_id=4614) then ratefinal='N/A';
    else if Rate_Calculated_by_SAS.sum=. then
ratefinal='NOT REPORTED';
    else do;
call
define('ratefinal',"style","style={tagattr='format:0.00
formula:(RC[-4]/RC[-1])*1000}");
```

Compute block for this column

```
        ratefinal=Rate_Calculated_by_SAS.sum ;  
call define( _col_,  
            "style/merge", "style=[VERTICALALIGN=MIDDLE]");  
*vertical alignment = centre;  
call define( _col_,  
            "style/merge", "style=[TEXTALIGN=CENTER]");  
*horizontal alignment = centre;  
end;  
endcomp;
```

Define statement for column with conditionally coloured cells

```
define percentile / display noprint;
```

```
define pctl / display 'Rate
```

```
Above*80th*Percentile*Among*Hospital*Gro  
up' style=test1;
```

Compute block for this column

```
compute pctl;  
  if pctl='G' then do;  
    pctl="";  
    call define(_col_,"style","STYLE=  
      [BACKGROUND=LIGHT GREY]");  
  end;  
endcomp;
```

Other proc report code used

- There was other code for the other columns used in the report.

Other proc report code used

```
rbreak after / summarize;
```

```
compute after;
```

```
  hospital_site='ONTARIO';  
endcomp;
```

```
run;
```

```
quit;
```

```
ods tagsets.excelxp close;
```

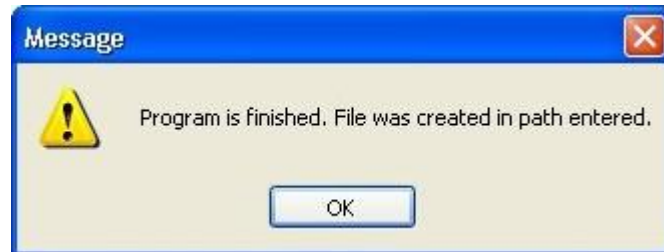
Code at the end of the macro

```
dm "POSTMESSAGE 'Program is finished. File was  
    created in path entered.'";  
data; run;quit;  
;
```

Code at the end of the macro

```
/*this is the goto point if there are errors above*/  
%finish: %if &syserr ne 0 %then %do;  
    %put ERROR: Macro Ended with Errors.;  
    dm "POSTMESSAGE 'ERROR: Macro Ended with  
    Errors.'";  
    data; run; quit;  
%end;  
ods listing;  
  
%mend cdi_print_excel;
```

Popup box indicates to user that the program is complete



Final Excel File

In Print Preview Mode

INFECTION Reporting Period: JAN 2011

Facility Number	Institution Number	Hospital Corporation	Hospital Site Name	Cases Reporting Facility	Cases Other Facility	Cases Unknown Source	Patient Days	Calc. Rate	Rate Above 80th Percentile Among Hospital Group	Outbreak Status	DHA Hospital Type	LHIN Name	LHIN Code	City
100	2345	Hospital Corporation 1	Hospital Site 1	N/A	N/A	N/A	N/A	N/A			Independent Health Facility	LHIN 21	21	CITY 1
101	2346	Hospital Corporation 2	Hospital Site 2	1	0	0	1,216	0.82			Small Community	LHIN 24	24	CITY 2
102	2347	Hospital Corporation 3	Hospital Site 3	0	0	0	253	0.00			Small Community	LHIN 16	16	CITY 3
103	2348	Hospital Corporation 4	Hospital Site 4	0	1	0	1,096	0.00			Small Community	LHIN 22	22	CITY 4
104	2349	Hospital Corporation 5	Hospital Site 5	0	0	0	1,338	0.00			Small Community	LHIN 25	25	CITY 5
105	2350	Hospital Corporation 6	Hospital Site 6	0	0	0	1,151	0.00			Small Community	LHIN 25	25	CITY 6
106	2351	Hospital Corporation 7	Hospital Site 7	0	0	0	325	0.00			Small Community	LHIN 28	28	CITY 7
107	2352	Hospital Corporation 8	Hospital Site 8	0	0	0	8,506	0.00			Mental Health	LHIN 18	18	CITY 8
108	2353	Hospital Corporation 9	Hospital Site 9	1	1	1	8,072	0.12		Yes - ended Jan 1, 2011	Large Community	LHIN 26	26	CITY 9
109	2354	Hospital Corporation 10	Hospital Site 10	0	0	0	934	0.00			Small Community	LHIN 27	27	CITY 10
110	2355	Hospital Corporation 11	Hospital Site 11	0	0	0	7,832	0.00			Ccc & Rehab	LHIN 21	21	CITY 11
111	2356	Hospital Corporation 12	Hospital Site 12	1	1	0	2,001	0.50			Large Community	LHIN 24	24	CITY 12
112	2357	Hospital Corporation 13	Hospital Site 13	0	0	0	1,391	0.00			Large Community	LHIN 24	24	CITY 13
113	2358	Hospital Corporation 14	Hospital Site 14	NOT REPORTED	NOT REPORTED	NOT REPORTED	NOT REPORTED	NOT REPORTED			Small Community	LHIN 23	23	CITY 14
114	2359	Hospital Corporation 15	Hospital Site 15	0	0	0	518	0.00			Small Community	LHIN 25	25	CITY 15
115	2360	Hospital Corporation 16	Hospital Site 16	0	0	0	835	0.00			Small Community	LHIN 27	27	CITY 16
116	2361	Hospital Corporation 17	Hospital Site 17	8	0	3	11,922	0.30			Large Community	LHIN 22	22	CITY 17
117	2362	Hospital Corporation 18	Hospital Site 18	N/A	N/A	N/A	N/A	N/A			Large Community	LHIN 22	22	CITY 18
118	2363	Hospital Corporation 19	Hospital Site 19	0	0	0	343	0.00			Small Community	LHIN 16	16	CITY 19
119	2364	Hospital Corporation 20	Hospital Site 20	0	0	0	1,699	0.00			Small Community	LHIN 27	27	CITY 20
120	2365	Hospital Corporation 21	Hospital Site 21	1	0	0	2,130	0.47			Large Community	LHIN 26	26	CITY 21
121	2366	Hospital Corporation 22	Hospital Site 22	0	0	0	1,744	0.00			Ccc & Rehab	LHIN 25	25	CITY 22
122	2367	Hospital Corporation 23	Hospital Site 23	0	0	0	445	0.00			Small Community	LHIN 25	25	CITY 23
123	2368	Hospital Corporation 24	Hospital Site 24	0	0	0	744	0.00			Small Community	LHIN 28	28	CITY 24
124	2369	Hospital Corporation 25	Hospital Site 25	2	0	1	1,097	1.82			Small Community	LHIN 18	18	CITY 25
125	2370	Hospital Corporation 26	Hospital Site 26	0	0	2	1,396	0.00			Small Community	LHIN 27	27	CITY 26
126	2371	Hospital Corporation 27	Hospital Site 27	0	0	0	2,981	0.00			Mental Health	LHIN 25	25	CITY 27
127	2372	Hospital Corporation 28	Hospital Site 28	0	0	0	5,726	0.00			Mental Health	LHIN 25	25	CITY 28

Final File – Up Close

	A	B	C	D
	Facility Number	Institution Number	Hospital Corporation	Hospital Site Name
1				
2	100	2345	Hospital Corporation 1	Hospital Site 1
3	101	2346	Hospital Corporation 2	Hospital Site 2
4	102	2347	Hospital Corporation 3	Hospital Site 3
5	103	2348	Hospital Corporation 4	Hospital Site 4

Final File – Up Close

	A	B	C	D	
	E	F	G	H	I
	Cases Reporting Facility	Cases Other Facility	Cases Unknown Source	Patient Days	Calc. Rate
1					
2	N/A	N/A	N/A	N/A	N/A
3	1	0	0	1,216	0.82
4	0	0	0	253	0.00
5	0	1	0	1,096	0.00

