

```

*ProcessBody ;
/*filename _webout "<path>/htmlout.html" ;*/

%let inputname = <name of input file> ;
%let standard = <name of accompanying standard> ;

/* Parameters for Section D of report (character columns to
review frequency) */

%let num = 3 /* how many columns to include in the table
report must equal the list below */
%let col1 = <column name> ;
%let col2 = <column name> ;
%let col3 = <column name> ;
/* however many columns set in let num = should have a let colN
statement */

libname sasin BASE "<path>" ;

libname fmcts BASE "<path>" ;

%let path      = <path>/includes/ ;
%let pathscore = <path>/programs/ ;

%let _ODSDEST=TAGSETS.HTMLPANEL;
%let _odsoptions = rs=none;
%let _odsstyle = &style ;
%let _ODSSTYLESHEET= ;

%let embedded_titles = yes ;
%let panelborder = 16 ;

ODS ESCAPECHAR= '^' ;

options device=activex fmtsearch=(fmcts.formats) ; /* libname
fmcts must be set */

%STPBEGIN ;
ODS ESCAPECHAR= '^' ;

data work.metrics ;
length metric_id $ 16 rowfailed rows_processed 8 ;

```

```

set sasin.&inputname. end=end ;                                /* libname
sasin must be set */
keep metric_id rowfailed rows_processed ;
rows_processed + 1 ;

/*First Rule      XXXXXXXX001 (e.g. Check NULL/Blank Value) */

***** XXXXXXXXXXXX001 *****

%let metric_id = XXXXXXXXXXXX001 ;
%let col      = <numeric column name> ;
%include "&path.numcolnull.sas" / source2 ; /* this calls
template for number column is null */

/*Second Rule     XXXXXXXX002 (e.g. Check NULL/Blank Value for
character column ) */

***** XXXXXXXXXXXX002 *****

%let metric_id = XXXXXXXXXXXX002 ;
%let col      = <character column name> ;
%include "&path.charcolnull.sas" / source2 ; /* this calls
template for character column is null */

/*Third Rule      XXXXXXXX003 (e.g. Check character column to ensure
it is on reference list ) */
%let metric_id = XXXXXXXX003 ;
%let col      = <character column>;
%let fmttest   = $<character format name>. ; /* this format
would have been previously created from reference files */
%include "&path.charformat.sas" /source2 ; /* this calls the
template to check that a character column is found within a
format */

.... Add as many rules here as are defined in the standards
spreadsheet ......

run ;

```

```

/* this SQL summarizes the file above into one record per metric
(rule) */
proc sql ;
create table work.&inputname._metrics as
this table may or may not be permanent /* */
select metric_id ,
      sum(rowfailed) as rows_failed format=comma15.0 ,
      max(rows_processed) as rows_processed
format=comma15.0 ,
      sum(rowfailed) / max(rows_processed) as percent_failed
format=6.4
from work.metrics
group by metric_id
;
quit ;

%include "&pathscore.scorereport.sas" /source2 ; /* this calls
the score report */

%stpend ;

/* scorecard report starts here */

%let metricfile = sasin.&inputname._metrics      ;

proc sort data=work.&inputname._metrics
;
by Metric_ID ;
run ;

proc sort data=sasin.&standard
      out=work.&standard
;
by Metric_ID ;
where metric_id NE "" ;
run ;

```

```

data work.&inputname._scored    ;
merge work.&inputname._metrics (in=m)
      work.&standard.      (in=s)
      ;
by Metric_ID  ;
length Score 8 ;
if m ;

machine = "http://111.222.333.444:8080/" ;
engine  = "SASStoredProcess/do?" ;
progpath = "_program=SBIP://METASERVER%2F<folder>%2F<folder>
%2F<folder>%2F" ;
program  = "ReviewOfInputData" ;
paramA   = "colname=" ;
odsstyle = "&_odsstyle." ;
;

htmlvar = "<a href=" ||
"'" || /* beginning single quote for href */
trim(machine) ||
trim(engine) ||
trim(progpath) ||
trim(program) || '&' ||
trim(paramA)  || strip(Data_field_name)  ||
'&_odsstyle='    || strip(odsstyle) ||
"'>" ;

Metric_ID_linked = htmlvar || strip(Metric_ID) || '</a>' ;

Based_On = "Percent" ;

if      Based_On = "Percent" then Measure = Percent_Failed *
100;
else if Based_On = "Rows"     then do ;
      Measure = Rows_Failed ;
      if red_end = 999999999 then red_end =
Rows_Processed ;
      end ;
if green_beg <= red_end   then      /* Lower is Better */
do ;
/*           Lower Better          */
      if      Green_beg <= Measure <= Green_end  then Score
= 1 ;           /*      0 <= x <= 50      */
      else if  Green_end <  Measure <= Blue_end  then Score

```

```

= 2 ;          /*      50 < x <= 90      */
    else if Blue_end < Measure <= Red_end then Score
= 3 ;          /*      90 < x <= 100     */
    else
= 0 ;
    end ;
else                                /* Higher is Better */

do ;
/*      Higher Better      */
    if      Green_beg >= Measure >= Green_end then Score
= 1 ;      /*      100 >= x >= 90      */
    else if Green_end > Measure >= Blue_end then Score
= 2 ;      /*      90 > x >= 50      */
    else if Blue_end > Measure >= Red_end then Score
= 3 ;      /*      50 > x >= 0       */
    else
Score =
4 ;
    end ;
label Metric_ID_linked      = "Column Name"
      Metric_ID           = "Metric_ID"
      Score                = "Score"
      Rows_Failed         = "Rows Failed"
      Rows_Processed      = "Rows Processed"
      Percent_Failed      = "Percent Failed"
      ;
run ;

proc sort data=work.&inputname._scored  ;
by Metric_id data_field_name  ;
run ;

proc format ;
  value score
    1 = "Pass"
    2 = "Check"
    3 = "Fail"
    ;
  value colorsc    /* This format assigns colors to cells */
    1 = 'Light Green'
    2 = 'Cyan'
    3 = 'Very Light Red'
    ;
run ;

Proc sql ;

```

```

create table work.filesummary as
select
    Score ,
    put(score,score.) as ScoreLabel ,
    count(distinct Data_field_name) as Columns ,
    count(distinct Metric_ID) as Metrics
from work.&inputname._scored
group by 1 , 2
;

create table work.scoresummary as
select data_field_name ,
    Score ,
    count( distinct Metric_ID) as Metrics
from work.&inputname._scored
group by 1, 2
;
quit ;

proc contents data=sasin.&inputname. noplayout out=work.contents ;
run ;

proc sql noplayout ;
create table work.contents2 as
select memname ,
    sum(length) as record_length format=comma20.0,
    max(varnum) as number_variables format=comma15.0,
    max(nobs) as Number_records format=comma20.0 ,
    max(crdtate) as date_created format=datetimetime25.
from work.contents
group by memname
;
select put(datepart(max(crdtate)),worddate28.) into :filedate /*  

use of into with : creates a macro variable within SQL */
from work.contents
;
select put(timepart(max(crdtate)),timeampm12.) into :filetime
from work.contents
;
select memname into :datasetname
from work.contents
quit ;

/* This creates section A of the Report */

```

```

%let panelcolumns = 1 ; /* Tells ODS how many panels to use
 */

ods &_ODSDEST event=panel(start);

data test ;
length text $ 255 ;
text = "Report for %UPCASE(&inputname.) for &client." ;
output ;
run ;

title1 j=center '^S={preimage="http://<path to logo>"}' ;
title2 j=left height=14pt color=gray "Report Created on:" 
      j=right height=14pt color=gray "SAS Data:
&datasetname." ;
title3
      j=left height=12pt color=gray "%TRIM(%QSYFUNC(DATE(),
worddate28.)) at %TRIM(%SYFUNC(TIME(), TIMEAMPM12.))"
      j=right height=12pt color=gray "&filedate. at
&filetime." ;

proc report data=test nowd noheader style(report)={rules=none
frame=void }
      style(column)={font_weight=bold font_size=20pt just=center
foreground=blue /*background=white*/ } ;
run;

ods &_ODSDEST event=panel(finish);

%let panelcolumns = 2 ;

ods &_ODSDEST event=panel(start);

title ;
title2 h=12pt color=blue
      "File Header Information" ;
footnote ;

PROC TABULATE
DATA=WORK.CONTENTS2 ;
      VAR record_length number_variables Number_records
date_created;
      CLASS MEMNAME / ORDER=UNFORMATTED MISSING ;
      TABLE

```

```

record_length      = "Record Length"          *f=comma20.0
number_variables   = "Number of Variables"    *f=comma15.0
Number_records     = "Number of Records"      *f=comma20.0
date_created       = "Date Created"
*f=datetime25.

,
      Sum = " " *
      MEMNAME = "SAS Dataset Name"           ;
;

RUN;

title2 h=12pt color=blue
      "Summary Statistics Report" ;
footnote ;
proc print data=work.filesummary label noobs ;
var score / style={background=colors.} /* This applies the
color format */
var Columns metrics ;
sum columns metrics ;
format score score. ;
label Score = "Score"
      Columns = "Number of Columns"
      Metrics = "Number of Metrics"
      ;
format columns
      metrics   comma20.0
      ;
run ;

ods &_ODSDEST event=panel(finish);

/* This creates section B of the Report */

%let panelcolumns = 1  ;

ods &_ODSDEST event=panel(start);

title2 h=12pt color=blue
      "Column Summary Report" ;
footnote ;

proc print data=work.scoresummary label noobs ;
var data_field_name ;
var      score / style={background=colors.} /* This applies

```

```

the color format */
var   metrics ;
sum metrics ;
label Data_field_name = "Field Name"

      Metrics = "Number of Metrics"
      ;
format
      metrics   comma20.0
      score score.
      ;
run ;

ods &_ODSDEST event=panel(finish);

/* This creates section C of the Report */

%let panelcolumns = 1 ;
ods &_ODSDEST event=panel(start);

title2 h=12pt color=blue
      "Column and Metric Details Report" ;
footnote ;

proc print data= work.&inputname._scored label noobs ;
var   metric_id_linked data_field_name metric_label Rows_Failed
Rows_Processed Percent_Failed ;
var score / style={background=colors.}  ;
format score score.
      percent_failed percent6.1
      ;
label Data_field_name = "Column Name"
      metric_label      = "Metric Description"
;
run ;

ods &_ODSDEST event=panel(finish);

/* This creates section D of the Report */

%let panelcolumns = 2 ;
ods &_ODSDEST event=panel(start);

%macro Freq(num,top)  ;

```

```

data work.&inputname. ;
set sasin.&inputname. ;
keep
  %do I= 1 %to &num. ;
    &&col&I.. &&col&I.._len
  %end ;
;
%do I = 1 %to &num. ;
  &&col&I.._len = length(&&col&I..) ;
%end ;
run ;

proc freq data=work.&inputname. noplay order=freq ;
  %do I = 1 %to &num. ;
    table &&col&I.. / missing out=work.F_&&col&I.. ;
      table &&col&I.._len / missing
  out=work.F_&&col&I.._len ;
  %end ;
run ;

%do I = 1 %to &num. ;
  Title1 "Top &top. Frequency for Values of &&col&i.." ;
  proc print data=work.F_&&col&I.. (obs=&top.) ;
  run ;
  Title1 "Top &top. Frequency for Length of &&col&i.." ;
  proc print data=work.F_&&col&I.._len (obs=&top.) ;
  run ;
%end ;

%mend ;

%Freq(&num.,&top.) ;

ods &_ODSDEST event=panel(finish);

```