ABSTRACT

The SAS® System provides a wide array of reporting tools. The SAS/ASSIST report button and the REPORT procedure offer both ease of use and powerful functionality. The information delivery strategies described here include exception reporting, drill down, and links to graphical charts using the REPORT procedure.

INTRODUCTION

Human Resources departments traditionally demand much from report writing tools. Both the volume and the variety of reports the department must produce are large. Therefore, the reporting needs of a Human Resources department and of a Human Resources Information System (HRIS) are rich ground for examples of how well the SAS System can respond.

The following discussion begins with tools that are immediately accessible to the end user. The SAS/ASSIST report button opens the door to point-and-click interfaces with which to create both simple and complex reports. The discussion moves on to presentation strategies that deliver information more effectively. PROC REPORT enables the end user to view information from different perspectives to assess alternatives and explore implications. Higher levels of integration and productivity can be achieved by embedding the functionality of PROC REPORT in a SAS/AP software application to create an end-user reporting environment.

THE SAS/ASSIST REPORT BUTTON

The SAS System offers direct access to reporting tools with the SAS/ASSIST report button. There is no need to request help from an MIS department. You can generate the information you need immediately. Among the tools available are a tool that produces tabular reports and a tool that helps you design your own ad hoc report.

U.S. government regulations require Human Resources departments to spend a lot of time submitting government reports. One agency that needs annual reports of the racial and ethnic breakdown of the work force is the Equal Employment Opportunity Commission (EEOC). Tabular summary reports are particularly appropriate to display these frequency counts organized into rows for each EEOC job category and columns for each racial group.
Display 3 Enhanced EEOC report

If your report is not as well-suited to the TABULATE procedure, you might prefer to use the SAS/ASSIST button for designing your own report. A salary forecast report requires grouping observations into departmental categories, computing a variety of salary increases, and displaying grand totals. The prompting facility of the REPORT procedure guides you through initial variable selection and grouping. Display 4 shows the DEPART variable defined for grouping.

Display 4 Prompter Windows Defining the DEPART Variable

Other PROC REPORT windows enable you to define computed columns and add grand totals.

Display 5 COMPUTE Window for the PLUS4 Computed Column

The inequities suggested by this report could result from men being paid more to do the same jobs as women, or the information could mean that only men are getting the jobs with higher pay. A second report displays the mean salary for men and women by job code.
Display 9 Job Code by Sex with Default PMENUs

An Information Delivery Solution

Extracting information from independent reports can be awkward. A better solution offers an integrated set of reports with links that a user can follow for alternative perspectives on the same information. The tools used in this example are reports displayed in PROC REPORT, specialized PMENUs created using PROC PMENU and displayed in the REPORT window, and bar charts created using PROC GCHART.

Display 10 Salary by Sex with Salary PMENUs

Display a bar chart of salaries in the Administration department by selecting Administration in the Department column, pulling down NewView, and selecting Departmental Chart...

ANOTHER HRIS PROBLEM: ADMINISTRATION OF LEAVE POLICY

Another area of concern to the Human Resources department is responsible use of leave benefits. A review of companywide leave records begins with a summary that includes the total sick days and vacation days for each employee. A second report displays more detail for each separate occasion of leave with accompanying comments if necessary. Note the effect of the FLOW option in the COMMENTS column.
An Information Delivery Solution

It would be more convenient to link these two reports so that details are readily accessible and comments easy to add. The tools needed to implement the solution are reports displayed in PROC REPORT, specialized PMENUs for the REPORT window, and the FSEDIT procedure.

To drill down from the employee summary to a detailed report select an employee name, pull down the Drill menu (Display 15), and select Drill down. Return to the summary report by selecting Drill up.

| Display 13 | Summary Leave Report with Default PMENUs |

| Display 14 | Detailed Leave Report with FLOW Option |

| Display 15 | Summary Leave Report with Drill PMENU |

Add a comment by selecting the Comments field to the right of the leave event, pulling down the Note menu, selecting the Add a comment... dialog, and pressing the OK button. If a comment already exists PROC REPORT invokes PROC FSEDIT on the observation containing the comment. If no comment exists PROC REPORT invokes FSEDIT on the comments data set and you must add a new observation. Comment changes appear the next time you generate this report.

HRS REPORTING NEEDS

The literature on Human Resource Information Systems as well as experience at SAS Institute indicates that reporting is particularly important in this arena. Commonly expressed needs of an HRIS reporting environment include

- selecting from a menu of standard reports
- designing and saving customized reports
- building and reviewing reports interactively
- sorting on any number of columns—including hidden columns
- summarizing and subtotalling using both sums and means
- subsetting the data
- distinguishing exceptions and outliers visually.

The examples you have seen testify to the ability of PROC REPORT to deliver most of this functionality. You can embed PROC REPORT in a SAS/AF application to realize even more advantages.
Display 18 SAS/AF HRIS Application Interface

PROC SQL created data views that bring together information from relationally organized SAS data sets. The public and private libraries are collections of report definitions. The public library contains standard reports. The private library contains customized reports saved by the user of the application. Pressing either button displays a list of the available report definitions. Buttons along the bottom of the window enable you to display a report in the REPORT window, print a report, edit an existing report definition to save in a private library, or design a new report. In each case the code submitted by the application invokes PROC REPORT with the appropriate options specified.

SUMMARY OF PROC REPORT FEATURES

The following features of PROC REPORT play important roles in the examples presented in this paper:

- the prompt facility
- report definitions
- the RLOAD command of PROC REPORT
- the CALL DEFINE routine and the EXECUTE command of PROC REPORT
- report profiles.

The Prompt Facility

The SAS/ASSIST Design a report button and the Design button of the HRIS application both invoke PROC REPORT with the PROMPT option specified. The prompt facility guides you through the process of selecting variables and setting up the report. This facility is an aid to the novice user but can also be a convenient interface for experienced users.

Report Definitions

Create a report definition using either PROC REPORT windows and the RSTORE command or PROC REPORT statements and the OUTREPT option. Once the definition exists you can refer to it using the REPORT – procedure statement option, as follows:

```sas
proc report nowd data=hris.leavevw
   report='hris.sugi92.leaveum.rept';
run;
```

You can also refer to it using the RLOAD command (see the next section). Report definitions reside in catalog entries with the .REPT extension. They cannot be edited directly, but they can be converted to PROC REPORT statements using the LIST procedure statement option or the RESOURCE command of PROC REPORT. From the SOURCE window you can save the statements to a .SOURCE catalog object or to an external file.

The RLOAD Command

Use the RLOAD command to apply another report definition to the current data set or view. The previous examples use the RLOAD command to implement links between reports including drill down to a report with more detail.

```sas
rload hris.sugi92.leaveu;where name='Smith, John'
```

CALL DEFINE Routine and EXECUTE Command

The CALL DEFINE routine of PROC REPORT defines attributes associated with a particular row and column of the report. The arguments of CALL DEFINE are:

- column a column name or number
- attribute an attribute name such as COLOR, FORMAT, or COMMAND
- value the value of the attribute such as RED, COMM9, or RLOAD

The CALL DEFINE routine always appears in a COMPUTE block. If the COMPUTE block is attached to a report column, then it executes once for each row of the report. If the COMPUTE block is attached to a BREAK then it executes on the BREAK rows of the report. In either case the row for which CALL DEFINE specifies an attribute is the currently active row of the report. You can execute CALL DEFINE conditionally to handle exception reporting as follows:

```sas
if percent > 20 & percent < 30
   then call define( ...col_, 'color', 'yellow );
else if percent > 30
   then call define( ...col_, 'color', 'red' );
```

Use the EXECUTE command to submit commands attached to a report row and column by the COMMAND attribute of CALL DEFINE. In applications it may be desirable to use PMENUs that replace the EXECUTE command with some phrase relevant to the application environment. Drill down in the previous example is implemented by the following CALL DEFINE statement and a pull-down menu that submits EXECUTE when the user selects Drill down.

```sas
call define( ...col_, 'command',
   'rload hris.sugi92.leaveu;where name='||trim(name)||'"');
```

Report Profiles

A report profile is a catalog entry named REPORT.PROFILE that contains information about how PROC REPORT executes. The profile contains:

- color information for most of the PROC REPORT windows,
- flags to indicate that the default execution mode is WINDOWS, NOWINDOWS, or PROMPT, and that the RKEYS window should or should not be displayed,
- and the location of PMENU catalog entries to use in the REPORT and COMPUTE windows.

The last information figures importantly in the HRIS examples. Three items specify related information.
• The PROFILE= option in the PROC REPORT statement names a catalog that contains a REPORT.PROFILE entry.

• The REPORT.PROFILE entry contains the name of a library and the name of a catalog in that library that contains a .PMENU entry to use in the REPORT window.

• The .PMENU entry contains the pull-down menus created by PROC PMENU.

Figure 1 illustrates the relationships among these three items.

proc report profile=plibname.pcatalog ...
+--PLIBNAME.PCATALOG.REPORT.PROFILE--+
| Pull-down menu library: PMENUlib |
| Pull-down menu catalog: PMENUcat |
| REPORT window PMENU: XXXX |
+------------------------------------------+

Figure 1 Locating a REPORT.PROFILE Entry

Create the REPORT.PROFILE catalog entry using the PROFILE command in PROC REPORT. When you press the OK button in the PROFILE window, PROC REPORT saves a REPORT.PROFILE entry in your SASUSER.PROFILE catalog. If REPORT.PROFILE remains in your SASUSER.PROFILE catalog, PROC REPORT uses the profile information whenever you invoke the procedure. To control use of profile information with the PROFILE= option move REPORT.PROFILE to some other catalog that can be referred to in the procedure statement.

NONINTERACTIVE EXECUTION OF PROC REPORT

PROC REPORT is designed to be used either as an interactive tool or as a non-interactive (or batch) tool. There are statements to use with PROC REPORT that enable you to define any report that you can define interactively. The following statements define the employee leave summary report used previously.

proc report data=hris.leavevw headline;
column depart ansalary plus4 plus6 plus8;
define depart / group format:$14. "Department" ;
define ansalary / mean format=dollar12.2 "Average Annual Salary" ;
define plus4 / computed format=dollar12.2 "Plus 4" ;
define plus6 / computed format=dollar12.2 "Plus 6" ;
define plus8 / computed format=dollar12.2 "Plus 8" ;
compute plus4 ;
  plus4 = ansalary + .04 ;
endcomp ;
compute plus6 ;
  plus6 = ansalary + .06 ;
endcomp ;
compute plus8 ;
  plus8 = ansalary + .08 ;
endcomp ;
break after / dol summarize ;
endcomp ;
compute pctdif ;
  pctdif = (_eL - _eL1) / _eL ;
endcomp ;
/* Use color and reverse video to mark differences that are particularly bad. */
absdif = abs( pctdif ) ;
endcomp ;
proc report data=hris.salaryvw headline;
column depart ansalary, sex pct:tdif ;
define depart / group format:S14. "Department" ;
define ansalary / mean format=dollar14.2 "Average Annual Salary" ;
define sex / across format:$sexfmt. width .. ,4 "By Sex"-- ;
define pctdif / computed format=percent10.2 "Percent Difference" ;
compute depart / char length"'.!! ;
  * Find the name of the graphics segment that contains */
  * the departmental bar chart */
  if depart=' Administration' then name=" ADMIN" ;
  else if depart=' Development' then name=" DEV" ;
  else if depart=' KIS' then name=" MIS" ;
  else if depart=' QA' then name=" QA" ;
  else if depart=' Sales' then name=" SALES" ;
  else if depart=' Service' then name=" SERVICE" ;
  /* Associate a GRAPH1 command with each department */
  if name = " " then do ;
    site = " command" ;
    call define( _col_, attr, 'graph1 hris.sugi92.'trim(name)".gseg" ) ;
  end ;
  endcomp ;
compute pctdif ;
  /* Column 2 contains values for men. Column 3 contains values */
  /* for women. Calculate the percentage difference between the */
  /* two. */
  if _c2 != 0 & _c3 != 0 then do ;
    pdiff = (_c2 - _c3) / _c2 ;
  /* Use color and reverse video to mark differences that are particularly bad. */
  absdif = abs( pctdif ) ;
endcomp ;
```
color = ""
if absdif > .1 and absdif <= .2
then do;
    color = 'yellow'; reverse = 0;
end;
else if absdif > .3 and absdif <= .4
then do;
    color = 'yellow'; reverse = 1;
end;
else if absdif > .4
then do;
    color = 'red'; reverse = 0;
end;
else if absdif > .5 and absdif <= .6
then do;
    color = 'red'; reverse = 1;
end;
else if absdif > .6
then do;
    color = 'red'; reverse = 1;
end;
/* Mark the report value */
if color = "" then do;
call define(_col_ , 'color', color);
if reverse then call define(_col_, 'rvsvideo', 1);
end;
end;
endcomp;
end after / do summarize;

Job Codes by Sex
proc report data=hris.salaryvw headline;
column jobcode ansalary, sex pctdif;
define jobcode / group format=S4. wJobCode;
define ansalary / mean format=dollar14.2 "Average Annual salary";
define sex / across format=Ssexfmt. width=14 HBy Sex/ __ H;
define pctdif / computed format=percentl0.2 "Percent Difference"
compute pctdif;
/* Column 2 contains values for men. Column 3 contains values for women. Calculate the percentage difference between the two.
if _c1 <> _c3 then do;
pctdif = (_c3 - _c1) / _c3;
/* Use color and reverse video to mark differences */
/* that are particularly bad. */
if absdif > .1 and absdif <= .2
then do;
    color = 'yellow'; reverse = 0;
end;
else if absdif > .3 and absdif <= .4
then do;
    color = 'yellow'; reverse = 1;
end;
else if absdif > .4
then do;
    color = 'red'; reverse = 0;
end;
else if absdif > .5 and absdif <= .6
then do;
    color = 'red'; reverse = 1;
end;
else if absdif > .6
then do;
    color = 'red'; reverse = 1;
end;
/* Mark the report value */
if color = "" then do;
call define(_col_ , 'color', color);
if reverse then call define(_col_, 'rvsvideo', 1);
end;
end;
endcomp;
end after / do summarize;
```

### Job Codes with Ranges
```
proc report data=hris.salaryvw headline;
column jobcode position depart n ansalary.(min max);
define jobcode / group format=S6. wWidth="Job Code";
define position / group format=$32. width=19 "Job Title";
define depart / group format=-14. ftDepartments";
define n / format=5.0 "Count";
define ansalary / sum format=DOLLAR14.2 "Annual Salary";
```

### Detailed Leave Report
```
proc report data=hris.leavevw headline;
column name eno depart start type dur comment;
define name / order format=$32. vidth=1 "Name of Employee";
define eno / order noprint;
define depart / order noprint;
define start / display format=date. "Start Date";
define type / across format=S8. "Type of Leave";
define dur / sum format=DOLLAR14.2 "Minimum";
define max / format=DOLLAR1. "Maximum";
```

### Salary Report PROC PMENU code
```
proc pmenu cat=salary.reports;
```

### Menus
- **File** menu: menu=file;
- **Search** menu: menu=search;
- **NewView** menu: menu=newview;
- **Help** menu: menu=help;

### Print
- **Print** menu: item "Set form name" dialog-setform;
- **Set print file** dialog-setprint;
- **End**;
- **Quit** dialog-quit;
```
dialog setform 'formname il';
text 12 all 'Specify a FORM entry to control printed output.';
text .3 ill 'Leave blank to display the name of the current form.';
text *6 all 'Enter form name: ';

dialog setprint 'PRTFILE il';
text 12 all 'Specify the fileref or pathname of a print file.';
text ,3 ill 'Leave blank to display the current print file.';
text 15 illi '{enclose path name in quotes}';

dialog quit 'QUIT';
text ,2 OJq 'Are you sure you want to quit PROC REPORT?';

menu search;
item 'Where' dialog=where;
item 'Where also' dialog=wherealso;
item 'Undo last where' selection=wherundo;

dialog where 'WHERE il';
text 12 all 'Enter where clause: ';

dialog wheralso 'WHERE ALSO il';
text 12 all 'Enter where also clause: ';

selection wherundo 'WHERE UNDO';

menu locals;
item 'Report Options' selection=reportoptions;
item 'List' selection=list;

menu drill;
item 'Drill down' selection=drilldown;
item 'Drill up' selection=drillup;

selection execute 'EXECUTE';
selection end 'END';

menu note;
item 'ADD a comment' dialog=comment;
item 'EDIT a SAS data set' dialog=dataset;

dialog comment 'EXECUTE';
text 12 all 'If the COMMENTS field for the leave dates of interest is selected press OK. Otherwise, press CANCEL and select the appropriate COMMENTS field.';

dialog dataset 'FSEDIT ;;il';
text 12 &q 'Library: ';
text 2 &q len=8 color=yellow;
text 13 &q 'Member: ';
text 3 &q len=8 color=yellow;
text 14 &q '{four level screen name}: ';
text 15 &q len=10 color=yellow;

menu help;
item 'PROC REPORT' selection=rephelp;
item 'Keys';

selection rephelp 'Help';
run;

The SAS/AF HRIS Application

**** SOURCE ****

/* Define application libraries. */
if libname('hris', 'sasuser') then msg = 'sysmsg';
if libname('salary', 'sasuser') then msg = 'sysmsg';
if libname('leave', 'sasuser') then msg = 'sysmsg';

buttons = ' ';
libname = ' ';
catname = ' ';
repname = ' ';
return;

main:
/* Handle the Public and Private buttons by retrieving */
/* a report definition name from the public or the private */
/* library. If the user pushed the design button remove */
/* all report name information. */
if buttons = 'Exit' then do;
_status_ = 'W';
return;
end;
else if buttons = 'Public' then do;
libname = 'hris';
catname = 'sugi92';
repsname = 'SU9i92';
repname = {catlist: 'hris.sugi92', 'rep': 1, 'W'};
pubname = repname;
prvname = repname;
end;
else if buttons = 'Private' then do;
libname = 'sasuser';
catname = 'hris';
repsname = 'HRIS';
pubname = repname;
end;
else if buttons = 'Open' then do;
libname = repname catlist: 'sasuser.hriss, rep', 'N';
pubname = prvname;
end;
else if buttons = 'Design' then do;
libname = repname;
catname = pubname;
prvname = repname;
call execfunc('end');
return;
end;
else if buttons = 'Show' then do;
libname = repname;
catname = pubname;
prvname = repname;
call execfunc('end');
return;
end;
else if buttons = 'Exit' then _status_ = 'R';
return;
end;

term:
if _status_ = 'C' then do;
_status_ = 'H';
return;
end;

if buttons = 'Exit' then _status_ = 'R';

/* the data set selection determines the view and profile */
/* needed by PROC REPORT. */
if d1 = ' ' then do;
dname = 'hris.pwftvw';
profile = 'profile-hris.public';
missing = 'missing';
end;
else if d1 = ' ' then do;
dname = 'hris.pwftvw';
profile = 'profile-hris.public';
end;
else if d1 = ' ' then do;
dname = 'hris.personal';
profile = 'profile-hris.public';
end;
else if d1 = ' ' then do;
dname = 'hris.personal';
profile = 'profile-hris.public';
end;
else if d1 = ' ' then do;
dname = 'hris.personal';
profile = 'profile-hris.public';
end;
else if d1 = ' ' then do;
dname = 'hris.personal';
profile = 'profile-hris.public';
end;
else if d1 = ' ' then do;
dname = 'hris.personal';
profile = 'profile-hris.public';
end;
else if d1 = ' ' then do;
dname = 'hris.personal';
profile = 'profile-hris.public';
end;
else if d1 = ' ' then do;
dname = 'hris.personal';
profile = 'profile-hris.public';
end;
else if d1 = ' ' then do;
dname = 'hris.personal';
profile = 'profile-hris.public';
end;
else if d1 = ' ' then do;
dname = 'hris.personal';
profile = 'profile-hris.public';
end;
else if d1 = ' ' then do;
dname = 'hris.personal';
profile = 'profile-hris.public';
end;

 SAS, SAS/AF, and SAS/ASSIST are registered trademarks or trademarks of SAS Institute Inc., Cary, NC, USA.

761