

A Generalized Macro-Based Data Reporting System to Produce Both HTML and Text Files

Jeff F. Sun, Blue Cross Blue Shield of North Carolina, Durham, North Carolina

Abstract

This paper will address the inter-connection between SAS System and Inter-/intra-net technologies with a macro-based reporting system. This system is a generalized macro-based data reporting system, which can be used to create customized HTML format files for Web browsers and text format files for hard copies. This system was initially built in the PC environment based on a 3M system structure, which can make it easy to build, update, and maintain the whole system.

The production of text files follows the regular process of external file creation in the SAS System. The creation of HTML files uses DATA `_NULL_` and HyperText Markup Language to produce the HTML 3.2 tag set supported by Netscape or Microsoft Internet Explorer or other Web browsers. Three obvious advantages of using this system can be seen as follows. First of all, this reporting system provides us a tool to extend our information delivery ability with SAS System. In addition this system significantly reduces the time required to produce tables from the SAS data sets for both text and HTML format files. Finally, this system can be used to easily and timely update and maintain tables or lists or other reports in a Web format.

I. Introduction

With a rapid development of information delivery across the internet or within an intranet and an exponentially increasing interest in the internet and web-based technologies, it is widely thought that web-based forms can be an ideal, and often superior substitute for paper forms

used to collect and report data for the SAS System. This inter-connection between inter-/intra-net technologies and the SAS system provides us the ability to process and analyze data, exchange information between sources at different locations, and display the outputs of our applications.

For the current SAS reporting systems or applications, the files are generally created as text files or hard-copies for data reporting and further analysis. In order to integrate SAS system with internet related technologies for the collecting, processing, analyzing, and reporting up-to-date, error-free, logically correct, and meaningful data, it is obviously good to extend our reporting system to produce both text and HTML files. In order to quickly create HTML format files and easily update these files when the new data sets are available, it is necessary for us to develop an application which can use the SAS system to process the data and output both text and HTML format files.

This paper will introduce a **Generalized Macro-based *REPORT*ing** system --- `%GMREPORT()`. `%GMREPORT()` is a powerful and flexible SAS macro-based reporting system which can produce either text files for hard copies or HTML files for internet browser display from simple SAS macro calls. It enables a user to easily and quickly produce or modify or update summary tables and customize the table layout. `%GMREPORT()` can also produce a pre-formatted SAS output presentation of data. This paper doesn't intend to discuss the line or page control for report writing since the topic on how the SAS system can be used to create HTML files is much more interesting at this time.

The paper is organized as follows. In section I, closing here, a brief introduction to the generalized macro-based data reporting system %GMREPORT() is presented. In section II a system structure of this macro-based data reporting system is discussed. The creation of three kinds of files, i.e. text, pre-formatted HTML, and HTML files, are detailed in section III. The next section contributes a user interface design. Some concluding remarks follow in section V.

II. System Structure

The programming structure of this macro-based reporting system is based on a 3M structure (Sun, 1997) for easy code building, system maintenance, and macro access. Figure 1 shows a 3M programming system structure. It consists of one linkage macro, three operation macros, one utility macro,¹ and one drive program.

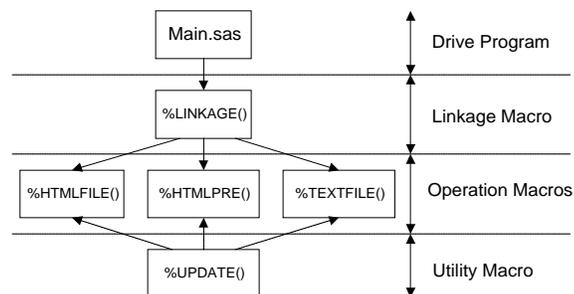


Figure 1. Flowchart of the System Structure

There are three major operation macros built for both text file and HTML file creation. They are %TEXTFILE(), %HTMLFILE(), and %HTMLPRE(). %TEXTFILE() produces a plain text file just like the regular production of text files. %HTMLFILE() will give an HTML format text file based on the layout of browser. %HTMLPRE() creates a pre-format HTML file, whatever you have from the SAS printout, you can create exactly the same format output for the browser.

¹ In general, a macro-based reporting system could have more than one utility macro.

The HTML file creation uses DATA _NULL_ and Hypertext Markup Language (HTML 3.2) to produce HTML format files, which can be displayed using any internet browsers. The DATA _NULL_ step concatenates all input data sets, uses conditional data processing to write rows and columns in the report, and produces the HTML 3.2 tag set supported by Netscape or Microsoft Internet Explorer or other internet browsers in the report.

All these macros are linked together by a linkage macro %LINKAGE(), which calls each operation macro conditionally based on the specification from the user. The linkage macro %LINKAGE() uses the %IF-%THEN statements to make a selection of macro execution from these operation macros (see Figure 2).

```

%MACRO LINKAGE(FILETYPE=);
%IF &FILETYPE=HTMLFILE %THEN
%DO;
%HTMLFILE()
%END;
%ELSE %IF &FILETYPE=TEXTFILE %THEN
%DO;
%TEXTFILE()
%END;
%ELSE %IF &FILETYPE=HTMLPRE %THEN
%DO;
%HTMLPRE()
%END;
%MEND LINKAGE;
  
```

Figure 2. Logic for the Linkage Macro %LINKAGE()

Utility macros are built for updating macro variables in the linkage macro and operation macros. In this example, there is only one utility macro, i.e. %UPDATE(), created for updating operation macros. One of the significant advantages for using modular utility macros is that changes made in one place will update all the macro values in all programs.

This paper will concentrate on the SAS programming logic of operation macros, that is, how to create HTML files from a DATA _NULL_ step in SAS system.

III. File Creation

a. %TEXTFILE(): Regular Text File Creation

Even though the Web provides a way to create Web pages in new and exciting ways, some readers still need many things off-line for the business. This requires that the system should perform the regular text file creation as well. The production of text files in this system is just like the regular text file creation.

The regular text file is created using DATA _NULL_ to output the data needed and table structure, formatting, and headings. Figure 3 shows a printout of a sample data report created by the operation macro %TEXTFILE().

Quarter	Month	CHILD		YPO		PCP		Total
		Regular	Stude	Regular	Stude	Regular	Stude	
1st Quarter	January	181784	25288	30673	3543	8252	1600	854106
	February	175844	24263	32473	3814	8608	16758	826725
	March	170239	23215	32485	3595	8999	17114	828756
2nd Quarter	April	188877	24188	31411	3425	10267	1793	833238
	May	136470	24184	31184	3780	10458	18075	822419
	June	150847	24012	28980	4047	10558	18488	813926
3rd Quarter	July	149293	24362	31745	3383	10923	1833	828108
	August	115477	23223	30823	3682	10847	18179	820428
	September	138530	23049	27242	4621	11807	20458	824078
4th Quarter	October	144141	23812	26615	3345	11238	20758	820058
	November	113864	23088	23990	3911	11304	22134	820951
	December	124736	24112	26088	4181	11887	22788	827478

Figure 3. A Text File Created by %TEXTFILE()

In order to convert this file, i.e. the exact same page format, to an HTML file, one can use either HTML tag sets for table creation or HTML tag set for pre-formatted output. In the next two parts of this section, the HTML file creation using SAS system will be discussed in detail.

b. %HTMLFILE(): HTML File Creation

HTML is the markup language used to create the documents displayed in internet browsers. An HTML document consists of text and tags that mark the structure of the document. The HTML file creation is based on the DATA _NULL_. One can create an HTML tag set for the internet browser to display information with the internet format and use a filename statement

to assign a *fileref* to an external file for storing the HTML file. Figure 4 illustrates a printout of an HTML file when viewed with an internet browser. Tables consist of three basic parts: the caption, headings, and rows of cells. One assembles HTML tables cell by cell, one horizontal row at a time. Because HTML defines tables on a row-by-row basis,² it could sometimes be difficult to keep track of the columns. The printout of the SAS text files is definitely useful to make a sketch of the table so one knows what the title, sub-title, heads, and footnote, and the values of each cell are.

Figure 4. An HTML File Created by %HTMLFILE()

Some SAS programming techniques for creating this kind of HTML files can be highlighted as follows.

(1) Caption --- Title and Subtitles;

The creation of a title and subtitles with HTML 3.2 is very simple. Table captions tell the reader what the table is. One can use <CAPTION> tag to add title and sub-title to the existing table (see Figure 5). Tools to process HTML files could extract them into a separate file, or automatically number them, or treat them in special ways simply because they are captions.

```
PAGETOP:
  PUT @2 '<HTML>/'
  @4 '<TITLE>/'
  @6 'SAMPLE TABLE FOR SUGI 23/'
  @4 '</TITLE>/'
```

² This is consistent with report writing using DATA _NULL_.

```

@6 '<CENTER>'/
@6 "<CAPTION ALIGN=CENTER> &TITLE <BR>"/
@6 "&SUBTITLE </CAPTION></CENTER>"/
@6 '<BR><P>'/
@6 '<BODY BGCOLOR=WHITE>'/
@8 '<CENTER>'/
@8 '<TABLE BORDER=1 COLOR=BLACK>'/
@10 '<FONT COLOR=BLACK>';
RETURN;

```

Figure 5. Creating Title and Sub-Title

It is recommended that a **<CAPTION>** tag for a table should include the PAGENTON portion, which creates HTML beginning setups. Please note that when macro variables are referenced in strings, they must be enclosed in double quotes.

(2) Footnotes;

For the table reports, it is necessary to have footnotes. A footnote can be simply produced using the paragraph tag **<P>** to start a new paragraph for the footnote (see Figure 6).

```

BOTTOM:
PUT @10 '</FONT>'/
@8 '</TABLE>'/
@8 '<BR><P><CENTER>'/
@8 "&FOOTNOTE </CENTER>"/
@6 '</BODY>'/
@2 '</HTML>';
RETURN;

```

Figure 6. Creating A Footnote

(3) Headings --- Column Names;

In order to produce an appropriate column name for each column, one can put the copy he wants to appear in each heading cell between the **<TH>** and **</TH>** tags. Think of TH as meaning table heading for each column.

```

IF _N_=1 THEN DO;
PUT @10 '<TR>'/
@12 '<TH COLSPAN=2>'/
@12 '<TH COLSPAN=2>CMM'/
@12 '<TH COLSPAN=2>PPO'/
@12 '<TH COLSPAN=2>PCP'/
@12 '<TH>'/
@10 '</TR>'/
;
PUT @10 '<TR>'/
@12 '<TH COLOR=BLACK> QUARTER'/
@12 '<TH> MONTH'/
@12 '<TH> REGULAR'/
@12 '<TH> STATE'/

```

```

@12 '<TH> REGULAR'/
@12 '<TH> SELECT'/
@12 '<TH> REGULAR'/
@12 '<TH> STATE'/
@12 '<TH> TOTAL'/
@10 '</TR>'/
;
END;

```

Figure 7. Spanning Columns

(4) Column Span;

Cells can span more than one vertical column or horizontal row. This makes information easier to read. Span columns when one wants to group cells under a single category or heading. The column span can be built by using **CLOSPAN** argument with the heading cell tag **<TH>**.

(5) Row Span;

In a similar way, one can create row span for the category variables. Span rows when information in a cell applies to several horizontal entries. There are two ways one can span a row with SAS system. Figure 8 shows row span with **ROWSPAN** argument.

```

IF FIRST.QRT THEN DO;
PUT @10 '<TR>'/
@12 '<TD ALIGN=CENTER ROWSPAN=3>' QRT /
@12 '<TD>' MONTH /
@12 '<TD ALIGN=RIGHT>' POP1 /
@12 '<TD ALIGN=RIGHT>' POP2 /
@12 '<TD ALIGN=RIGHT>' POP3 /
@12 '<TD ALIGN=RIGHT>' POP4 /
@12 '<TD ALIGN=RIGHT>' POP5 /
@12 '<TD ALIGN=RIGHT>' POP6 /
@12 '<TD ALIGN=RIGHT>' TOTAL /
@10 '</TR>'/
;
END;
ELSE DO;
PUT @10 '<TR>'/
@12 '<TD>' MONTH /
@12 '<TD ALIGN=RIGHT>' POP1 /
@12 '<TD ALIGN=RIGHT>' POP2 /
@12 '<TD ALIGN=RIGHT>' POP3 /
@12 '<TD ALIGN=RIGHT>' POP4 /
@12 '<TD ALIGN=RIGHT>' POP5 /
@12 '<TD ALIGN=RIGHT>' POP6 /
@12 '<TD ALIGN=RIGHT>' TOTAL /
@10 '</TR>';
END;

```

Figure 8. Spanning Rows with ROWSPAN

Figure 9 illustrates how to span rows without **ROWSPAN**.

```

IF FIRST.QRT THEN DO;
  PUT @10 '<TR>/'
    @12 '<TD ALIGN=RIGHT>' QRT /
    @12 '<TD>' MONTH /
    @12 '<TD ALIGN=RIGHT>' POP1 /
    @12 '<TD ALIGN=RIGHT>' POP2 /
    @12 '<TD ALIGN=RIGHT>' POP3 /
    @12 '<TD ALIGN=RIGHT>' POP4 /
    @12 '<TD ALIGN=RIGHT>' POP5 /
    @12 '<TD ALIGN=RIGHT>' POP6 /
    @12 '<TD ALIGN=RIGHT>' TOTAL /
  @10 '</TR>/'
;
END;
ELSE DO;
  PUT @12 '<TD>' /
    @12 '<TD>' MONTH /
    @12 '<TD ALIGN=RIGHT>' POP1 /
    @12 '<TD ALIGN=RIGHT>' POP2 /
    @12 '<TD ALIGN=RIGHT>' POP3 /
    @12 '<TD ALIGN=RIGHT>' POP4 /
    @12 '<TD ALIGN=RIGHT>' POP5 /
    @12 '<TD ALIGN=RIGHT>' POP6 /
    @12 '<TD ALIGN=RIGHT>' TOTAL /
  @10 '</TR>';
END;

```

Figure 9. Span Rows without ROWSPAN

c. *%HTMLPRE(): Pre-formatted HTML File Creation*

For some people, they prefer to have the SAS output format to be displayed on internet. Some people may review the same table report by an earlier version of browsers that didn't support tables.

A browser normally formats the text in an HTML file and ignores any extra spaces, tabs, or line returns which are added to the file. One can use pre-formatted text (the `<PRE>` tag) to line up his information in table-like columns - creating table without the table tags. The `<PRE>` formats the text based on the way specified. It retains all the spaces, tabs, and returns in the HTML file so that it looks exactly the same when viewed with a browser.

This task should be achievable without significant additional programming efforts. The SAS programming structure for this part is shown as Figure 10.

```

.....
DATA _NULL_;
  FILE HTMLPRE;

```

```

  PUT '<HTML>';
  PUT '<HEAD>';
  PUT '<TITLE>PRE-FORMATTED TEXT FROM
  SAS</TITLE>';
  PUT '</HEAD>';
  PUT '<BODY BGCOLOR=WHITE>';
  PUT '<FONT SIZE=4 COLOR=BLACK>';
  PUT '<PRE>';
  RUN;

PROC PRINTTO PRINT=HTMLPRE;
  RUN;

Regular DATA _NULL_;

PROC PRINTTO;
  RUN;

DATA _NULL_;
  FILE HTMLPRE MOD;
  PUT '</PRE>';
  PUT '</FONT>';
  PUT '</BODY>';
  PUT '</HTML>';
  RUN;

```

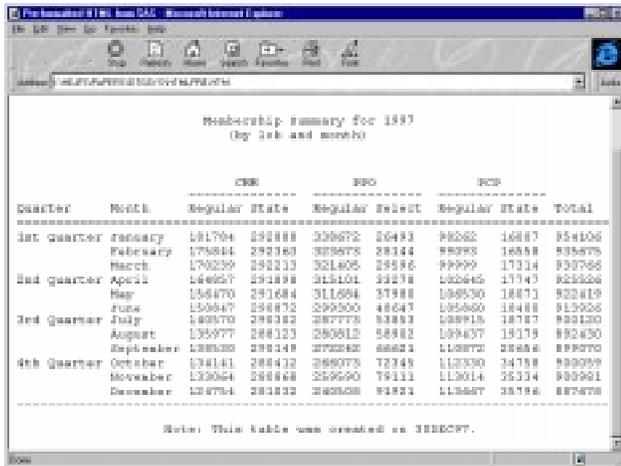
Figure 10. Coding Structure for %PREHTML()

There are three parts in the SAS programming for pre-formatted HTML file creation, i.e. beginning setup, body, and ending setup.

(1) **HTML Beginning Setup:** This part creates an HTML beginning setup and writes to an external file by using `DATA _NULL_`, for example, beginning tags `<HTML>`, `<HEAD>`, `<TITLE>`, `<BODY>`, `<PRE>`, etc.

(2) **Text Output:** In order to get a pre-formatted HTML file, one can redirect the SAS output to an HTML file using `PROC PRINTTO`. The `PRINTTO` procedure does not produce an output data set or printed output. With the `PRINTTO` procedure, one can specify a *fileref*, a filename, or a logical name as the destination for printed output or the SAS log. One can use `PROC PRINTTO` to route the print file to a permanent file.

(3) **HTML Ending Setup:** This part simply produces HTML ending tags and writes to the same external file. Figure 11 demonstrates the pre-formatted HTML file when viewed with an internet browser.



Quarter	Month	Regular Rate	Regular State	Regular Select	Regular Total	Total
1st Quarter	January	181794	282888	306678	26443	885803
	February	175844	282888	306678	28144	893658
	March	170239	282888	321408	28886	805461
2nd Quarter	April	166657	282888	315101	32278	797024
	May	158470	282888	311884	37988	791330
	June	150847	282888	299300	48247	781363
3rd Quarter	July	148370	282888	287778	53883	773029
	August	139977	282888	280912	58882	762659
	September	138288	282888	272842	66824	761042
4th Quarter	October	134141	282888	266078	72348	755455
	November	130064	282888	258280	79113	750345
	December	128784	282888	248280	84823	744875

Figure 11. A Pre-Formatted HTML File Created by %HTMLPRE()

IV. Parameter Specification

In the macro call, you can simply specify the parameter FILETYPE to get a customized format. Also it is possible to specify other parameters to define titles, or footnotes, or column names. For instance, the title definition is setup for the user specification. One can simply specify the macro variable TITLE as following:

TITLE = Membership Summary for 1997,

The title is default to the center position. The sub-title can be also setup in the same way.

SUBTITLE = (by month and lob),

The footnote for the table can be specified by the user as well.

FOOTNOTE = Note: This table was created on &sysdate%str(),

Please note that %str() is used to write period dot in the footnote.

A user can also specify the column span, or row span, or color, or font, or other setups. These will not be discussed here due to a limited space in this paper.

V. Conclusion

Several significant advantages for this reporting system can be seen as follows. First of all this reporting system provides us a tool to extend our information delivery ability with the SAS

system. Secondly, this macro-based data reporting system significantly reduces the time required to produce tables from SAS data sets for both text and HTML format files. In addition, using this system can update and maintain tables or lists or other reports in a Web format timely and easily. Finally the investment for developing this kind of reporting system can be considered small. It is now quite possible to develop a functional Web-based reporting system on a desktop computer using the SAS system with only a small financial investment and efforts.

For further work on this area, the development of a graphic user interface with SAS/FRAME and SAS/SCL could improve the current data reporting system and make it much easier to use.

Trademarks

SAS is a registered trademark of the SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are registered trademarks or trademarks of their respective companies.

Reference List

Sun, Jeff F., (1997) "On the Macro-Based Information", in *Proceedings of the Fifth Southeast Annual SAS User Group Conference*, Jacksonville, Florida.

Any questions or comments regarding the paper may be directed to the author:

Jeff F. Sun
 Health Services Information And Analysis
 Blue Cross Blue Shield of North Carolina
 ET 600, 3rd Floor
 Durham, NC 27705
 Phone: (919)-765-7854
 E-mail: jeff_sun@scp11.bcbsnc.com