

Sign of the Times: Using SAS® to Produce Conference Signage

Daniel K. Downing, Caremark, Scottsdale, Ariz.

ABSTRACT

"Sign, sign, everywhere a sign" Are you at the right place at the right time? Who knows? But if you checked at the door on the way in, you at least know you are at the correct place and time for the presentation you wanted to see. This paper will guide you through the processes used to convert the program schedule for the Western Users of SAS® Software conference into event signs that show the who, what, where and when of this gathering. It will describe a little graphics handling, a lot of SAS® data manipulation and some of the experimental SAS® 9 features found under the Output Delivery System (ODS) PDF destination. It will bring all of these techniques together, and show you how to produce your own "sign of the times."

INTRODUCTION

There are many logistical tasks that transpire along the way to conducting a successful conference. The site has been selected. The papers have been published. The rooms have been readied. All the sections have been scheduled and the presentations prepared. Once the conference begins, the focus shifts to making sure that attendees can easily find the location they are looking for. This is done quite nicely through the use of the program guide and the signs posted outside each room. These signs confirm what papers are being presented in that room at what time. They can be as simple as a white board with magic markers or as elaborate as a neon marquee. What these signs must do is clearly convey a concise view of the event, be flexible and fluid up to the start of the event (do not carve your signs in stone), and be easy to read as the attendee passes by. One obvious way to create accurate, dynamic and attractive signage is to automate the process. With more than 33 session signs needed for this conference, automation was more of a necessity than a luxury.

THE BACKGROUND IMAGE

The background image that is used for the signs is a JPG file. The beginning image is the conference logo. The next image has had the text removed. Finally, the contrast, brightness and gamma settings are adjusted to create a very subtle, watermark image. There are a variety of graphic editor programs that can be used to accomplish this. The image is then placed on a background containing a banner of blue (not shown). This will be used as the standard template for all the signs being constructed.



THE DATA

The data presented on the signs occurs at two distinct levels: section and paper. The section-level data, used to construct the sign's heading, consists of the section name and presentation date. The paper-level data, used to produce the detail in the sign's body, consists of the author(s) name, time and paper title. For the purpose of having a self-contained, executable example, the source code at the end of this paper uses a DATA step to build a demonstration data set. The variables used are shown below. The version of this program that was used to actually produce the signs has the schedule data imported from a Microsoft Excel® file.

```
data work.schedule ;
  length section_id          $5    section_name          $30
         presentation_date    $8    presentation_time    $8
         presentation_type    $13   presentation_length   $8
         presenter_1_first_name $11  presenter_1_last_name $17
         . . . . .
         presenter_4_first_name $9   presenter_4_last_name $6
         presentation_name     $159  presentation_room     $27 ;
run ;
```

THE PROCESS

Once the schedule data has been successfully read into a data set, it needs to be manipulated in order to be prepared for presentation. The three major steps needed to process the schedule data and produce signs are constructing the author and time variables, isolating the distinct section/day/sessions to drive the process and building a macro program containing the PROC REPORT code which will be executed to produce the signs.

BUILD AUTHOR AND TIME VARIABLES

Some data manipulation is required to transform all the elements into a clean, predictable and consistent format. It involves acknowledging and accounting for multiple authors and extrapolating the text date and time from the PRESENTATION_DATE, PRESENTATION_TIME and PRESENTATION_LENGTH variables. With SAS[®] 9, there is a new family of functions, affectionately referred to as the “feline” functions. They are used for the con”CAT”enation of variables, which does away with the need to use the TRIM and LEFT functions on the contributing components. This results in less code that is easier to understand. In this application, the CATX function is used for putting together the variable containing the names of multiple authors. Not only does the CATX function allow for the assignment of separation characters, it does not resolve them if there is no data element, thereby eliminating the need to go back and remove unnecessary separation characters.

OLD METHOD:

```
TEXT_AUTHORS = trim(left(PRESENTER_1_LAST_NAME)) || " / " ||
               trim(left(PRESENTER_2_LAST_NAME)) || " / " ||
               trim(left(PRESENTER_3_LAST_NAME)) || " / " ||
               trim(left(PRESENTER_4_LAST_NAME)) || " / " ;

END_INDEX    = index(TEXT_AUTHORS, ' / ');
TEXT_AUTHORS = substr(TEXT_AUTHORS, 1, END_INDEX) ;
```

NEW METHOD:

```
TEXT_AUTHORS = catx(' / ', PRESENTER_1_LAST_NAME, PRESENTER_2_LAST_NAME,
                   PRESENTER_3_LAST_NAME, PRESENTER_4_LAST_NAME) ;
```

In order to create the section-level data, the PRESENTATION_DATE and PRESENTATION_TIME variables are used. Since there is no session (a.m./p.m.) flag in the data, the morning or afternoon status is determined by the scheduled start time. The HOUR function is used to examine just the hour portion of the start time. Once the session has been determined, it is concatenated behind the day of the week that the paper is being presented. This is accomplished by using the PUT function with the DOWNAME9. (day of week name) format.

```
if hour(PRESENTATION_TIME) < 12 then
  TEXT_TOD=upcase(catx(' ', put(PRESENTATION_DATE, downame9.), 'MORNING '));
else
  TEXT_TOD=upcase(catx(' ', put(PRESENTATION_DATE, downame9.), 'AFTERNOON '));
```

In a similar fashion, no end time exists in the data, just the PRESENTATION_TIME and PRESENTATION_LENGTH variables. The end time is calculated in seconds, formatted into an a.m./p.m. time by using the TIMEAMP8. format and concatenated with the start time to produce a text string showing the start and finish time of the paper. Although it would work, in this instance the CATX function is not used, as it does not inherently contribute any additional clarity.

```
TEXT_TIME = put(timepart(PRESENTATION_TIME), timeampm8.) || ' - ' ||
             put(timepart(PRESENTATION_TIME+(PRESENTATION_LENGTH*60)), timeampm8.);
```

ISOLATE DISTINCT SECTION/DAY/SESSIONS

The second part of the process is to take the schedule data, distill down to the individual section/day/session and populate them into the macro variables which will be used to drive the macro program. Surfacing the unique values is accomplished through the use of PROC SORT with a NODUPKEYS option. Once those values have been established, a DATA _NULL_ step with a CALL SYMPUT is used to build the macro variables. The macro variable name is made up of a meaningful constant (SEC=Section) and an integer, representing the number of times through the data step (_n_) as part of the macro name. With no subsetting IF or WHERE code, this is in effect, the OBS number. `call symput('SEC'!!left(put(_n_, 5.)), SECTION_ID);` The macro variable built for the first observation will be named &SEC1 and will contain the value from SECTION_ID variable. This process is repeated for

the section and session values. Additionally, the total number of observations is captured for use as an index in the next step to control the DO loop.

BUILD MACRO PROGRAM TO EXECUTE PROC REPORT

The third step on the way to sign production is to construct the macro program that will execute the PROC REPORT code. Each section/day/session will produce an individual page in the PDF document. The macro program executes the PROC REPORT code for each unique occurrence captured in the previous step. The looping function of the %DO will be driven by the total number of observations. The PROC REPORT code is the place where your personal style can be expressed by implementing any of the `style=` options available when using ODS. These options provide control over several aspects of the output appearance such as font, color, size and style.

USING OUTPUT DELIVERY SYSTEM FEATURES

The final steps involved in producing the schedule signs are creating a customized style sheet that identifies the background image, setting the desired ODS options, and executing the macro program to produce the signs.

CREATE CUSTOMIZED STYLE SHEET

Custom style sheets allow modification of specific aspects of system provided existing style sheets. They start by inheriting all the properties of the parent and pick up new property assignments applied through PROC TEMPLATE. There are some experimental features in SAS® 9 that the background image can utilize. They require the use of an ODS escape character so that simple formatting instructions can be inserted into the ODS document after the escape character is encountered. The “^” character is frequently used for this purpose. The options that can be used to format the JPG image are SCALE, TILE and SCALETOFIT. The SCALE option will scale the background image to the size of the PDF document. The TILE option will create multiple repeating background images. The SCALETOFIT option can be used to preserve the aspect ratio of the background image.

```
ods escapechar='^';
proc template;
  define style SIGNS_01 ;
    parent=styles.printer;
    replace Body from Document "Controls the Body file." /
      backgroundimage="C:\WUSS2005logo_Skyline_Faded.jpg^scale";
  end;
run;
```

SETTING ODS OPTIONS AND EXECUTION

ODS offers many choices. However, at the basic level, ODS must be turned on, parameters set, procedures executed and then turned off. In this case the procedure that is executed is the macro program containing the PROC REPORT. When the source code included at the end of this paper is executed, the PDF output is written to the default path in the “start ln” or “current” folder of the session, which is usually displayed on the bottom, right side of the session. To view the results generated by running the code, open the “schedule.pdf” file.

CONCLUSION

While creating a background image in an HTML document has been available since the first release of ODS, having background images available in the PDF destination opens a new world of possibilities. This technique provides a simple solution for creating more attractive PDF documents that are data driven. It may be used for signs, as shown here, or for creating any PDF documents with a repeating template image.

ACKNOWLEDGMENTS

Thanks to all the Caremark SAS® users who allow me to help them meet their informational needs.

CONTACT INFORMATION

Comments and questions are valued and encouraged. Contact the author at:

Daniel K. Downing
Consulting IT Analyst
Caremark
9501 E. Shea Blvd.
Scottsdale AZ, 85260
Phone: 480.391.4282
Email: daniel.downing@caremark.com

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries.

Other brand and product names are trademarks of their respective companies.

THE SOURCE CODE

```
* CREATE DEMONSTRATION DATA SET * ;

data work.schedule ;
  length section_id          $5    section_name          $30
         presentation_date    8    presentation_time    8
         presentation_type    $13  presentation_length    8
         presenter_1_first_name $11 presenter_1_last_name  $17
         presenter_2_first_name $11 presenter_2_last_name  $17
         presenter_3_first_name $11 presenter_3_last_name  $17
         presenter_4_first_name $11 presenter_4_last_name  $17
         presentation_name     $159 presentation_room     $27 ;

SECTION_ID = 'CC' ;          SECTION_NAME = "Coders' Corner" ;
PRESENTATION_DATE = '21SEP2005'd ; PRESENTATION_TIME = '8:00:00't ;
PRESENTATION_TYPE = 'contributed' ; PRESENTATION_LENGTH = 10 ;
PRESENTER_1_FIRST_NAME = 'Art' ; PRESENTER_1_LAST_NAME = 'Pantier' ;
PRESENTER_2_FIRST_NAME = 'Bob' ; PRESENTER_2_LAST_NAME = 'White' ;
PRESENTER_3_FIRST_NAME = 'Dylan' ; PRESENTER_3_LAST_NAME = 'Bell' ;
PRESENTER_4_FIRST_NAME = 'Joanne' ; PRESENTER_4_LAST_NAME = 'Flack' ;
PRESENTATION_NAME = 'Show Down At The MACRO Corral' ;
PRESENTATION_ROOM = 'Main Ballroom A' ;
output ;

SECTION_ID = 'CC' ;          SECTION_NAME = "Coders' Corner" ;
PRESENTATION_DATE = '21SEP2005'd ; PRESENTATION_TIME = '8:30:00't ;
PRESENTATION_TYPE = 'contributed' ; PRESENTATION_LENGTH = 10 ;
PRESENTER_1_FIRST_NAME = 'James' ; PRESENTER_1_LAST_NAME = 'Bassman' ;
PRESENTER_2_FIRST_NAME = 'Sally' ; PRESENTER_2_LAST_NAME = 'Rand' ;
PRESENTER_3_FIRST_NAME = ' ' ; PRESENTER_3_LAST_NAME = ' ' ;
PRESENTER_4_FIRST_NAME = ' ' ; PRESENTER_4_LAST_NAME = ' ' ;
PRESENTATION_NAME = 'Getting The Most From Your SAS Investment' ;
output ;

SECTION_ID = 'CC' ;          SECTION_NAME = "Coders' Corner" ;
PRESENTATION_DATE = '22SEP2005'd ; PRESENTATION_TIME = '13:30:00't ;
PRESENTATION_TYPE = 'contributed' ; PRESENTATION_LENGTH = 10 ;
PRESENTER_1_FIRST_NAME = 'Tony' ; PRESENTER_1_LAST_NAME = 'Archer' ;
PRESENTER_2_FIRST_NAME = 'Mary' ; PRESENTER_2_LAST_NAME = 'Easter' ;
PRESENTER_3_FIRST_NAME = 'Paul' ; PRESENTER_3_LAST_NAME = 'Suffolk' ;
PRESENTER_4_FIRST_NAME = 'Becca' ; PRESENTER_4_LAST_NAME = 'Crudo' ;
PRESENTATION_NAME = 'Doing Even More With Even Less' ;
output ;

SECTION_ID = 'CC' ;          SECTION_NAME = "Coders' Corner" ;
PRESENTATION_DATE = '22SEP2005'd ; PRESENTATION_TIME = '13:45:00't ;
PRESENTATION_TYPE = 'contributed' ; PRESENTATION_LENGTH = 10 ;
PRESENTER_1_FIRST_NAME = 'Ray' ; PRESENTER_1_LAST_NAME = 'Joyner' ;
PRESENTER_2_FIRST_NAME = ' ' ; PRESENTER_2_LAST_NAME = ' ' ;
PRESENTER_3_FIRST_NAME = ' ' ; PRESENTER_3_LAST_NAME = ' ' ;
PRESENTER_4_FIRST_NAME = ' ' ; PRESENTER_4_LAST_NAME = ' ' ;
PRESENTATION_NAME = 'Running SAS At The Speed Of Light' ;
output ;

run ;

* PROCESS SCHEDULE TO BUILD COMBINED AUTHORS & TIME VARIABLES * ;
```

```

data SESSIONS (keep=SECTION_ID          TEXT_SECTION  TEXT_TOD  TEXT_TIME
                TEXT_PRESENTATION      TEXT_AUTHORS  PRESENTATION_DATE
                BEG_HOUR                TEXT_TIME_AUTHORS) ;

set SCHEDULE ;
length TEXT_AUTHORS $50  TEXT_TOD $ 25  TEXT_TIME_AUTHORS $ 100
        TEXT_PRESENTATION $ 200 ;

TEXT_AUTHORS = catx(' / ',PRESENTER_1_LAST_NAME,PRESENTER_2_LAST_NAME,
                  PRESENTER_3_LAST_NAME,PRESENTER_4_LAST_NAME) ;

TEXT_SECTION = put(SECTION_ID,$section.) ;
TEXT_PRESENTATION = PRESENTATION_NAME;
BEG_HOUR = hour(PRESENTATION_TIME) ;

if hour(PRESENTATION_TIME) < 12 then
    TEXT_TOD =
        upcase(catx(' ',put(PRESENTATION_DATE,downname9.),' MORNING ')) ;
else
    TEXT_TOD =
        upcase(catx(' ',put(PRESENTATION_DATE,downname9.),' AFTERNOON')) ;

TEXT_TIME = put(timepart(PRESENTATION_TIME),timeampm8.)||' - '||
            put(timepart(PRESENTATION_TIME +
(PRESENTATION_LENGTH*60)),timeampm8.);

TEXT_TIME_AUTHORS = TEXT_TIME||'          '||TEXT_AUTHORS ;

run ;

*   SORT SCHEDULE BY SECTION / DATE / TIME          * ;

proc sort data=SESSIONS ;
    by SECTION_ID PRESENTATION_DATE BEG_HOUR TEXT_TIME ;
run ;

*   CONSOLIDATE SCHEDULE TO GET 1 OBS PER SECTION / TIME OF DAY * ;

proc sort data=SESSIONS out=SESSIONS_UNIQUE
    (keep=SECTION_ID TEXT_SECTION TEXT_TOD) nodupkeys ;
    by SECTION_ID TEXT_SECTION PRESENTATION_DATE descending TEXT_TOD ;
run ;

*   CREATE MACRO VARS / TOTAL OBS - UNIQUE SECTION & TIME OF DAY * ;

data _null_;
    set SESSIONS_UNIQUE nobs=nobs;

    if _n_ = 1 then call symput('totobs',put(nobs,5.));

    call symput('SEC'!!left(put(_n_,5.)),SECTION_ID);
    call symput('TXT_SEC'!!left(put(_n_,5.)),TEXT_SECTION);
    call symput('TXT_TOD'!!left(put(_n_,5.)),TEXT_TOD);

run;

*   CREATE MACRO TO PRODUCE SIGNS BY SECTION / TIME OF DAY (AM/PM) * ;

%macro RUN_SIGNS ;

    %do i = 1 %to &totobs ;

```

```

PROC REPORT nowd data=SESSIONS
  style(report)=[bordercolor=white cellpadding=13 ];

  where TEXT_SECTION = "&&TXT_SEC&i" and
        TEXT_TOD      = "&&TXT_TOD&i" ;

  column TEXT_TIME_AUTHORS TEXT_PRESENTATION ;

  define TEXT_TIME_AUTHORS / group center " "
    style={foreground=cx004713
           cellwidth=2 in
           font_size=4
           font_style=italic
           font_weight=bold};

  define TEXT_PRESENTATION / group left " "
    style={foreground=cx004713
           cellwidth=5 in
           font_size=7
           font_weight=bold} ;

  title1 c=white h=25 bold j=c "WHITE FILL " ;
  title2 c=BLACK h=20 bold j=c "&&TXT_SEC&i" ;
  title3 c=BLUE h=15 bold j=c "&&TXT_TOD&i" ;
run;

%end ;

%mend RUN_SIGNS ;

* BUILD CUSTOMIZED STYLE THAT USES GRAPHIC IMAGE AS BACKGROUND * ;
* * * CHANGE THE PATH AND NAME TO USE -YOUR- GRAPHIC IMAGE * * * ;

ods escapechar='^';

proc template;

  define style SIGNS_01 ;
    parent=styles.printer;
    replace Body from Document "Controls the Body file." /
      backgroundimage=
        "C:\WUSS2005logo_Skyline_Faded.jpg^scale";

  end;

run;

* SET UP ODS DESTINATION AND EXECUTE MACRO TO PRODUCE SIGNS * ;

options nodate nonumber ;
ods listing close ;
ods pdf style = SIGNS_01
  file = "'SCHEDULE.pdf';" ;

%RUN_SIGNS ;

ods pdf close;
ods listing ;
options date number ;

*-----* ;

```