Paper 213-2012

Programming the Provider Previews: Extreme SAS® Reporting

Christianna Williams, Independent Consultant, Chapel Hill, NC
Louise S. Hadden, Abt Associates Inc., Cambridge, MA

ABSTRACT

Each month, we produce more than 15,000 data-driven reports (each report being three pages) for nursing homes, previewing their five-star ratings on various measures. These reports are produced entirely by SAS®, including a macro toggle for test or production mode and the collection of 15,000 PDF documents into a single multi-gigabyte zip file. The elements of the program include SAS macros, the DATA step, the REPORT procedure, and the PDF statement in the SAS Output Delivery System (ODS). Each PDF document is named to easily upload into specific electronic mailboxes for each nursing home facility.

INTRODUCTION

Since 1998, the U.S. Centers for Medicare and Medicaid Services (CMS) has maintained a website, Nursing Home Compare, which provides detailed quality information about every certified nursing home in the country. In December 2008, CMS greatly enhanced the usability of the website by adding an easy-to-understand 5-star rating. Each nursing home receives one to five stars based on performance in each of three key quality domains (health inspections, reported staffing levels, and quality measures derived from mandated assessments of resident health and well-being) plus an overall quality rating. Calculation of ratings requires integration of information from both facility and resident-level data sources. SAS® was used extensively in analysis to support the development of the rating system, and it is currently used to process data to refresh the ratings each month, based on newly collected data in each domain.

A preview of the month's ratings in the form of a customized two-page PDF report is generated for each provider every month. These reports are automatically loaded into providers' e-mailboxes by means of specific identifying information embedded in the file name for each provider's report. The data sources for the reports are the Provider Rating file created at the conclusion of data processing for the month, and a separate data base with MDS (Minimum Data Set of resident assessments) 3.0 data for as many quarters as are available (the two most recent are currently published in the reports). Specific variables used in the customization of the reports are: State, Provider Number, Facility ID, Provider Name, Overall Rating, Health Inspection Rating, QM Rating, Staffing Rating, RN Staffing Rating, and selected Quality Measures from MDS 3.0 at both the facility and national level. Additionally, if a provider's ratings put them at risk for being a "special focus facility", a warning paragraph is conditionally printed on the first page of the preview.

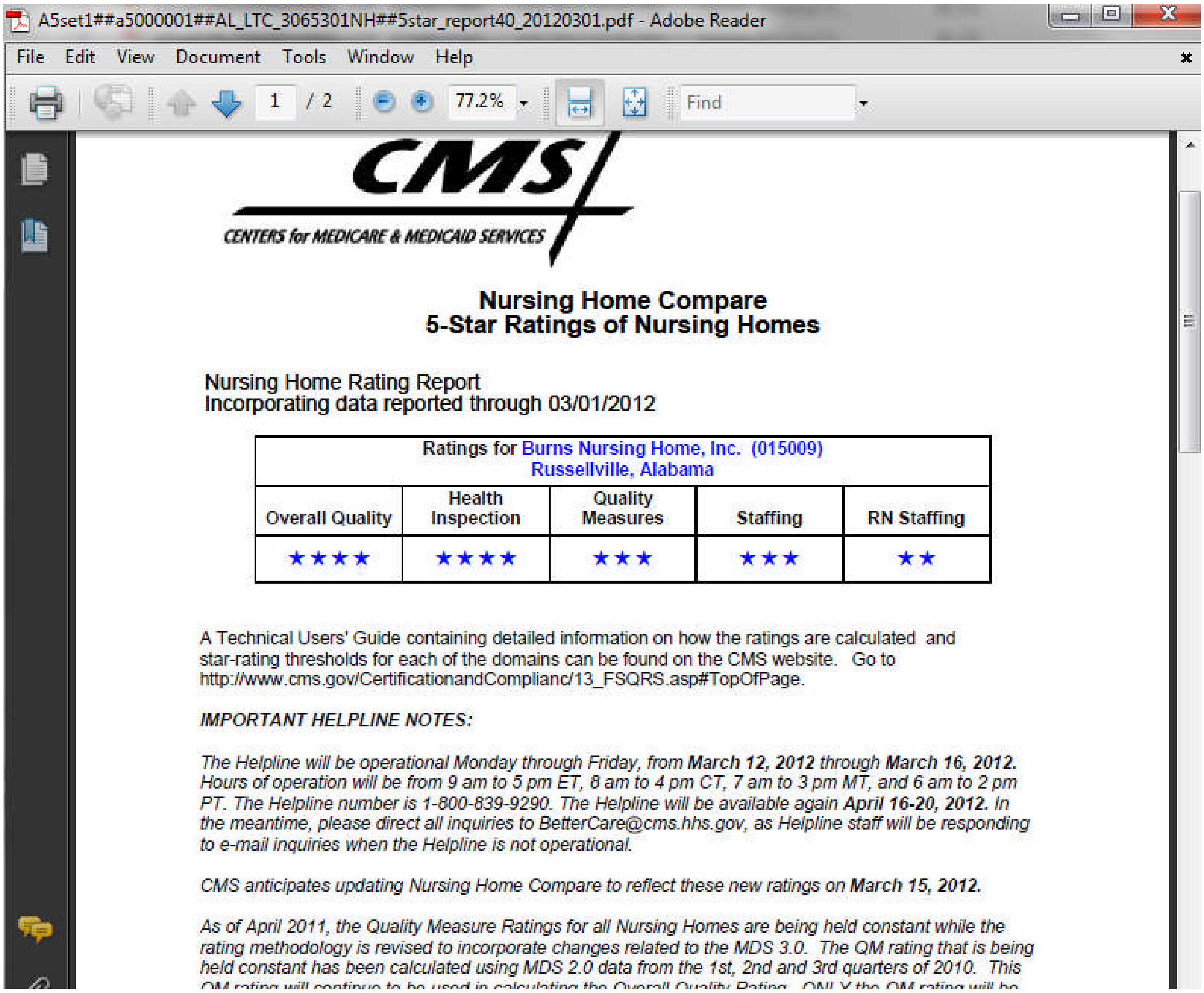
The provider previews, which were originally three pages in length and included extensive explanations of how the ratings were created, were recently revised to incorporate new information on MDS 3.0. This new data will eventually be published on the Nursing Home Compare website. The provider previews are now two pages in length, and refer the reader to a webpage which explains how the ratings are created rather than printing the details in the preview.

The report also contains information on the Helpline availability, which changes from month to month depending on which month in a quarter is being presented in the report. Thus, while much of the two page report is static, the ratings (in the form of one to five stars), complaint and MDS 3.0 data, and information on Helpline availability change from month to month. Helpline information is the same for all providers, while the ratings are provider-specific.

The PDF reports are automatically generated in a SAS program which contains a template for the static information and empty "shells" for the provider number, provider state, name, facility id and rating information. This program is edited each month to change the report number and month being reported on (which are part of the PDF file name), and to update the Helpline availability information. The SAS program pulls in the CMS logo at the top, and displays ratings as one to five stars. PDF files are automatically named so as to conform to conventions used to load files into providers' electronic mailboxes. At the completion of the generation of approximately 15,700 PDF reports ranging from 40KB to 100KB in size for each provider in the country, the SAS program then "zips" the PDF files into a single zip archive approximately 2.7GB in size for transfer using WINZIP and the WINZIP command line interface. The zip archive is then encrypted and loaded to the Abt Secure File Transfer server for transfer to CMS. (This server runs software provided by Ipswitch Software, which creates a “brandable”, browser-based and FIPS 140-2 compliant transfer system.)

The provider preview program is currently being run on a Windows x64 Server, using SAS 9.2 Maintenance Release 3. It takes approximately 3-8 hours to run, depending on the system load on our server. It is run exclusively in batch mode. There have been many updates to the program as new data elements are added: a substantial change generates another version number. Currently we are on the tenth version.

An example is shown below.


A5set1##a5000001##AL_LTC_3065301NH##5star_report40_20120301.pdf - Adobe Reader
File Edit View Document Tools Window Help
1 / 2
77.2%
Find
CMS
CENTERS for MEDICARE & MEDICAID SERVICES

Nursing Home Compare
5-Star Ratings of Nursing Homes

Nursing Home Rating Report
Incorporating data reported through 03/01/2012

Ratings for Burns Nursing Home, Inc. (015009) Russellville, Alabama				
Overall Quality	Health Inspection	Quality Measures	Staffing	RN Staffing
★★★★	★★★★	★★★	★★★	★★

A Technical Users' Guide containing detailed infomation on how the ratings are calculated and star-rating thresholds for each of the domains can be found on the CMS website. Go to http://www.cms.gov/CertificationandComplianc/13_FSQRS.asp#TopOfPage.

IMPORTANT HELPLINE NOTES:

The Helpline will be operational Monday through Friday, from ***March 12, 2012*** *through* ***March 16, 2012.*** *Hours of operation will be from 9 am to 5 pm ET, 8 am to 4 pm CT, 7 am to 3 pm MT, and 6 am to 2 pm PT. The Helpline number is 1-800-839-9290. The Helpline will be available again* ***April 16-20, 2012.*** *In the meantime, please direct all inquiries to BetterCare@cms.hhs.gov, as Helpline staff will be responding to e-mail inquiries when the Helpline is not operational.*

CMS anticipates updating Nursing Home Compare to reflect these new ratings on ***March 15, 2012.***

As of April 2011, the Quality Measure Ratings for all Nursing Homes are being held constant while the rating methodology is revised to incorporate changes related to the MDS 3.0. The QM rating that is being held constant has been calculated using MDS 2.0 data from the 1st, 2nd and 3rd quarters of 2010. This QM rating will continue to be used in calculating the Overall Quality Rating. ONLY the QM rating will be

A full sample program and sample data is available on www.sascommunity.org, at http://www.sascommunity.org/wiki/Programming_the_Provider_Previews:_Extreme_SAS_Reporting.

BEHIND THE SCENES

In order to create the effect of a personalized letter for each provider, it is essential to remove the cell borders present in most SAS output. In order to do this, we need to create a style template that we’ll call ‘noborder’. This style template is useful for many purposes.

```
/* the noborder template */
       proc template;
```

```
           define style styles.noborder;
           parent=styles.minimal;
           style Table /
           rules = NONE
           frame = VOID
           cellpadding = 0
           cellspacing = 0
           borderwidth = 0;
       end;

       run;
```

There may be (are) times when we want to have cell borders in the provider previews. SAS and ODS allow you to change styles within your ODS call. This is also handy if your company would like to have any tables or graphs include a logo or special background, but not any surrounding text. Within your ODS call, you simply specify a reset prior to the procedure you want to be output with a particular style:

```
/* open your ODS call */
ods pdf file = ".\facilityPDFs\
A5set1##a50000&fnum.##&facnode.5star_report&reportno._&fileyear.&filedate..pdf"
style=styles.minimal notoc author='Abt Associates Inc.' bookmarklist=hide
anchor='Table' compress=9;
ods escapechar='^';
ods pdf startpage=no;
ods proclabel '5 Star Ratings';
.
.
.
/* reset style */
ods pdf style=styles.noborder;
.
.
.
/* reset style */
ods pdf style=styles.minimal;
.
.
.
/* close your ODS call */
ods pdf close;
```

CREATING “PARAGRAPHS” IN SAS

SAS makes a very powerful and flexible desktop publisher. It is possible to construct very wide variables, so that paragraphs can be created by concatenating text fields or variables using the CATT function. We use the CATT function (which eliminates trailing blanks) so that we can string “sentences” together with a leading blank. We use the escapechar n function to create line feeds. In the example below, the paragraphs are created in a data step, but you could also import a file with existing fields. We label the variables using the “hidden dragon” (ALT+255), which suppresses the printing of the label or variable name. The paragraphs can be printed using PROC REPORT with a template without borders discussed above.

```
data overall_blurb;
    length blurb blurba blurbb blurb0 blurb0y blurb1 blurb2 blurb3 blurb6
           blurbx blurby blurbz $ 8128
           sent0 sent1
           warn0 warn1 warn2 warn3 warn4
           sentence1
           sentence2 sentence3
           sentence4 sentence5
           sentence6
           $ 500 blurb23 $ 8128
           sentence16 sentence17 sentence17a  $ 500;
    set provider_preview (where=(provnum="&provnum"));
```

```
sentence16="A Technical Users' Guide containing detailed information on how the
ratings are calculated";
sentence17="  and star-rating thresholds for each of the domains can be found on the
CMS website.";
sentence17a="
^S={url='http://www.cms.gov/CertificationandComplianc/13_FSQRS.asp#TopOfPage'} Go to
http://www.cms.gov/CertificationandComplianc/13_FSQRS.asp#TopOfPage.";

blurb23=catt(sentence16,sentence17,sentence17a);


/* warning paragraph for SFF */

warn0='^n^n^{style [fontweight=bold fontstyle=italic fontsize=9pt color=red]
*** IMPORTANT MESSAGE:  Your health inspection score currently makes you a candidate';
warn1=' for the Special Focus Facility Program.';
    .
    .
    .
    run;
```

In the code snippet above, you will note that we have incorporated a number of “in-line” style changes and some other ODS ESCAPECHAR functions such as line feeds, url links, etc. Most of these functions, but not all, are available in V9.1.3. All are available in V9.2 and above. In this case, we are using the ^ as our ODS ESCAPECHAR. ^n is a line feed. ^n^n produces two line feeds, and so on (note there is a new function to do this – newline, available in 9.2). In warn0, you will note that we explicitly change font color. Because we are stringing sentences together, it is important to do this because of style inheritance – whatever comes first in the “paragraph” holds unless explicitly assigned otherwise.

The paragraphs can be printed using PROC REPORT with a template without borders discussed above.

USING MACRO VARIABLES

In the code snippet shown above, we use a number of data driven macro variables which are set earlier in the program. Since macro variables resolve to text, it is easy to incorporate them into our report. Note that we ALSO find the length of each variable used before converting it to a macro variable, create another macro variable with the length, and set the length of the “value” macro variable with the “length” macro variable before creating the “value” macro variable. This prevents any unwanted blanks in our “sentences”. An example of producing a macro variable from a numeric variable follows: The process is similar for character variables.

```
/* create some provider specific macro variables */
data len;
    set provider_preview (where=(provnum="&provnum"));
    lens=length(fac_id)+9;
    call symput('lens',lens);
    lenp=length(trim(left(provname)))  +1 ;
    call symput('lenp',lenp);
    lenr=length(put(rn_staffing_rating,$starfmt.));
    call symput('lenr',lenr);
    lenc=length(city_state);
    call symput('lenc',lenc);
    .
    .
    .
    run;
```

and

```
data temp;
    length fac_node $ &lens. parenfac $ 8 prov_name $ &lenp. rnrate $ &lenr.
    prov_city $ &lenc.;
    set provider_preview (where=(provnum="&provnum"));
    prov_name=propcase(trim(left(provname)));
    if index(prov_name,'Llc')>0 then prov_name=tranwrd(prov_name,'Llc','LLC');
    prov_city=city_state;
    fac_node=cats(state,'_LTC_',fac_id,'##'); /* note revised from state_cd */
    parenfac=cats('(',provnum,')');
    rnrate=put(rn_staffing_rating,$starfmt.);
    .
    .
    .
    run;
```

We also use macro variables to set file dates, create file names, and toggle between test and production mode. These macro variables are not data-driven, and are set at the top of the program for ease of processing. This allows us to make recurring monthly changes once at the top of the program. These macro variables are identical to those used throughout the whole suite of programs in our monthly processing.

```
/* set the dates for the current round of processing */

%let filedate=0602;
%let fileyear=2011;
%let filemo=%substr(&filedate,1,2) ;
%let filedy=%substr(&filedate,3,2) ;
%let fileyear2=2011;
%let runmode=production; /* alternatives are test and production - case matters */
%let reportno=31; /* 31 is for June 2011 */
```

INCLUDING GRAPHICS

In most letters, it is customary to use letterhead and/or an organization's logo. Since the provider previews are being sent out by CMS, we wanted to insert the CMS logo at the top of the first page of the preview as shown in the example above. SAS provides a number of options for incorporating custom graphics into ODS output. Among these options is using ODS text fields with a "preimage" or "postimage" style command. We chose this option because it was easy to control the location and size of the CMS logo on the page in relation to many other items. An example of this usage, placed after the ODS call but before other ODS text calls and procedures, follows below.

```
ods text='^S={preimage="cmslogo.tif"}';
```

We also use SAS Unicode characters to produce the ratings stars in the previews. SAS Unicode characters are available in SAS 9.2 and above. Unicode symbols are translated by ODS into graphic characters. This allows the addition of useful characters such as Greek letters, etc. without changing fonts. The syntax is ^{Unicode xxxx} where ^ is the specified ODS ESCAPECHAR and xxxx is the 4-digit Unicode value. The Unicode value for a star is 2605.

We constructed a format containing one to five stars depending on facilities' ratings (code snippets shown below). Character variables representing the star versions of the ratings were constructed using put statements and the star format. These character variables were then used in a PROC REPORT to produce a table with stars. Note that we had two text values in the format as opposed to one to five stars. We reduced the font size for these text values so that they fit into the cell width allotted with style commands.

Creating a format with Unicode characters

```
proc format;
value $starfmt
'10'='^{Unicode 2605}'
'20'='^{Unicode 2605}^{Unicode 2605}'
'30'='^{Unicode 2605}^{Unicode 2605}^{Unicode 2605}'
'40'='^{Unicode 2605}^{Unicode 2605}^{Unicode 2605}^{Unicode 2605}'
'50'='^{Unicode 2605}^{Unicode 2605}^{Unicode 2605}^{Unicode 2605}^{Unicode 2605}'
'70'='Too New to Rate'
'90'='Data Not Available';
run;
```

Constructing the character variables containing Unicode stars

```
data rptline;
    length col1 col2 col3 col4 col5 col1a col2a col3a col4a col5a $ 200;
    set provider_preview (where=(provnum="&provnum"));
    col1a=put(overall_rating,$starfmt.);
    . . .
if overall_rating in('10','20','30','40','50') then col1=col1a;
if overall_rating in('70','90') then col1=catt('^{style [fontsize=11pt]',
col1a,'}');
. . .
    label col1='Overall Quality'
          col2='Health Inspection'
          col3='Quality Measures'
          col4='Staffing'
          col5='RN Staffing';
run;
```

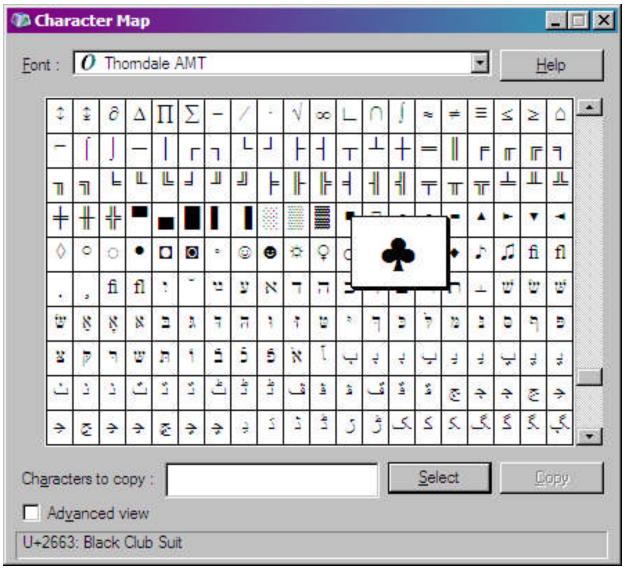
PROC REPORT using the star rating variables

```
proc report nowd data=rptline /* spacing=8 */
    style(report)=[cellpadding=3pt vjust=b]
    style(header)=[just=center font_face=Arial font_weight=bold font_size=10pt]
    style(lines)=[just=left font_face=Arial] split='|';
  columns ("Ratings for ^S={just=c font_size=10pt font_face=arial color=blue}&pname.
&parenfac.^n&provcity"
           col1 col2 col3 col4 col5);
  define col1 / style(COLUMN)={just=c font_face=Arial  foreground=blue
           font_size=12pt cellwidth=170 }
                      style(HEADER)={just=c font_face=Arial font_weight=bold
           font_size=10pt  };
  . . .

run;
```

Note that in this case, we change the font color within the PROC REPORT step and we use macro variables in a spanning header over the columns, modifying the font color of data dependent elements there as well.

To look up Unicode values on a system running Windows XP, open Start -> Programs -> Accessories -> System Tools -> Character Map. For example, I chose Thorndale AMT, which is used by SAS® V9.2. When a character on the screen is selected and/or hovered over, the Unicode VALUE is shown on the bottom of the screen (in this case the Black Club Suit is Unicode VALUE 2663.) As you can see, there are many other options than stars available, including bullets, etc.



Font : Thorndale AMT

Characters to copy :

□ Advanced view

U+2663: Black Club Suit

PRINTING PAGES WITH MULTIPLE PROCEDURE CALLS

Constructing “pages” in SAS is a challenge, for a number of reasons. The “rules” of pagination in each ODS tagset are quite complex and will not be addressed here. ODS RTF and ODS PDF both provide the ODS STARTPAGE command. The default within an ODS call in ODS PDF and ODS RTF is to generate a new “page” for each procedure. This default can be overridden by specifying ODS STARTPAGE=NO. Similarly, ODS STARTPAGE=YES will generate a new page. This capability allows us to stack up multiple procedures, ODS TEXT commands, etc. in a single page.

```
ods pdf startpage=no;

ods text='^S={preimage="cmslogo.tif"}';
. . .

ods pdf startpage=no;

proc report nowd data=rptline . . .

/* reset style */
ods pdf style=styles.noborder;
ods pdf startpage=no;

proc report nowd data=page1 . . .

run;

ods pdf startpage=yes;

proc report nowd data=page2 . . .
```

A very large caveat is that SAS will not “break” a “paragraph” (in this case, a paragraph is really a very long single variable so that does make sense.) It also is not as simple as counting lines when you are including graphics and multiple procedure calls on a single page. In our case, it is largely a process of testing and reviewing as the provider preview letters have evolved over time.

EFFICIENCIES USING MACRO PROCESSING AND PROC SQL

The frequent changes in the provider previews required us to build in as much flexibility into the program as possible, and to provide the option of running in “test” mode so that we can review each month’s previews stringently before running in production mode. At the top of the program, we create a test file with 15 providers of 4 different types: “normal” providers, providers with a warning paragraph, providers who are too new to rate, and providers with missing data. We set a macro toggle (also at the top of the program) to either “test” or “production”. In test mode, the small file of 15 providers is fed into the reporting macro. The resulting 15 provider previews are then zipped within the program into specific folder for review and archiving. These previews are reviewed, any necessary corrections are made, and then the program is run in production mode. In production mode, the file with all providers in the United States operating in a given month is fed into the reporting macro. All 15,700 or so provider previews are zipped within the program into a specific folder. In both test and production mode, the unit of observation (on which we report) is the provider – what goes into the macro call is the provider number. The different routines for test and production mode are conditionally executed based on a macro variable (set at the top of the program) RUNMODE.

Following the production run, at least 10 provider previews are reviewed from both the beginning and end of the run, to try and ensure the entire process completed successfully.

```
/* start of macro MAKEFIND which conditionally executes depending on macro variable
RUNMODE */

%macro makefind(infi);

data finder;
    set &infi;
run;
```

```
%mend makefind;

/* end of macro MAKEFIND */

/* conditionally execute macro MAKEFIND based on run mode to create finder file */

data _null_;
    if "&runmode"="test" then CALL EXECUTE('%makefind(test)');
    else if "&runmode"="production" then CALL EXECUTE('%makefind(finder)');
run;
```

We chose to use PROC SQL to accomplish the hard work of creating 15,700+ macro calls to the primary reporting macro RPTMAC. The driver macro shown below is a simple and elegant solution to this challenge. (Thanks to SAS-L!)

```
/* start of macro DRIVER to run macro RPTMAC for each record found in finder file
generated by macro MAKEFIND */

%macro driver ;

  proc sql noprint ;
     select provnum
     into :eid1-:eid999999
     from finder
     ;
     %let num_entities = &SQLOBS ;
  quit ;

  %do i = 1 %to &num_entities ;
   %let this_entity = &&eid&i ;
   %rptmac(&this_entity) ;
  %end ;

%mend driver ;
```

Remember that the finder file has been conditionally created from either our small test file or the full provider rating file for the month.

```
/* run macro DRIVER on finder file */

%driver;
```

MANAGING FILE SIZES

Although the source files used to create the provider previews are relatively small, 15,700+ PDFs can take up a fair amount of room (around 3GB). In addition, this program is run on a monthly basis, which could result in hundreds of gigabytes worth of provider previews. We utilize a number of techniques to help manage the size of each provider preview and to reduce the amount of space taken up by the provider previews in each month's processing folder.

While in-line style commands allow us to do great things, they also take up room. The information needed to "render" Unicode characters, font information, etc. is voluminous. PDFs, like MS Word documents, are larger than a flat text file because this information is built into the file. Interestingly, the font family matters to ODS PDF in its rendering of output. We had originally used the Arial font in designing the previews, but discovered that Arial and ODS PDF have a strained relationship. Using the font Helvetica instead of Arial reduces the size of each PDF by almost 50%. SAS also offers a compression option for ODS PDF. We use the highest compression option value in our ODS PDF call. For other similar, handy options for ODS PDF please see the ODS PDF tip sheet SAS puts out.

We also utilize the WINZIP Command Line Interface in conjunction with the X command in this program and others to zip up files within the program. In this case, we zip the PDF output generated at the end of the program. This zip file is encrypted with PGP, and transferred to CMS for distribution of the provider previews. The PDFs are deleted from the processing folder, and the zip file is archived on DVD, prior to our archiving of each month's processing (also accomplished in a zipping SAS program.) In the event that these files have not been deleted prior to the archiving run, we specifically exclude the folder containing the large provider preview zip file from our archive zip.

Many aspects of the provider preview program have evolved over time. Our zip processing is no exception. Originally, we had a hard-coded X command at the bottom of the program, with full path names (because we run exclusively in batch, we can use the . shortcut to denote the directory the program is running in.) The full path names had to be edited each month, which was problematic, especially since the zipping routine was at the bottom of a very long program. The solution to having to edit full pathnames was to use macro variables within the X command (see the excellent “Zipping Right Along” and “Zip and Email” papers in the references.)

Without macro variables in X command

```
/* start of macro TESTZIP to zip up test provider previews */

%macro testzip;
  x 'c:\progra~2\winzip\wzzip -a .\test_facilityPDFs\testpp.zip .\facilityPDFs\*.pdf'
%mend testzip;

/* end of macro TESTZIP */
```

With macro variables in X command

```
/* start of macro PRODZIP to zip up production run of provider previews */

%macro prodzip;
 X " ""c:\progra~2\winzip\wzzip""  -a
"".\facilityPDFs\A5set1##a5000001##5star_report&reportno..zip""
"".\facilityPDFs\*.pdf"" "
%mend prodzip;

/* end of macro PRODZIP */
```

Based on our macro toggle for production or test mode described above, we use CALL EXECUTE to conditionally execute our zip routines.

```
/* conditionally execute macros TESTZIP or PRODZIP based on run mode to zip up preview
files */

data _null_;
    if "&runmode"="test" then CALL EXECUTE('%testzip');
    else if "&runmode"="production" then CALL EXECUTE('%prodzip');
run;
```

THE FUTURE OF THE PROVIDER PREVIEWS

SAS 9.3 is out, with new defaults and greater graphics capacity. However, all the SAS tools described in the process of creating the provider previews are still relevant. We will still need to use the ability to construct paragraphs, issue inline style commands, manage file sizes, use Unicode characters, execute code conditionally, and take advantage of PROC SQL and Macro language efficiencies.

ACKNOWLEDGMENTS

Scott Huntley and Dan O’Connell of SAS have inspired the authors with great talks and classes on ODS PDF and ODS LAYOUT; and Cynthia Zender of SAS has always given us good advice on many and varied SAS explorations.

The authors gratefully acknowledges the contributions to the Nursing Home Compare 5-star project from:
Abt Associates: Allison Muma, Alan White and Michael Plotzke
Centers for Medicare and Medicaid Services (past and present): Edward Mortimore, Jonathan Freedlander, Daniel Andersen, Frank Nagy, Jenny Filopovits, Deme Umo and Thomas Kress.

REFERENCES AND RECOMMENDED READING

Gebhart, Eric. “Inline Formatting with ODS Markup.” Proceedings of SAS® Global Forum 2009 Conference. March 2009.

Hadden, Louise S. “Behind the Scenes with SAS®: Using Custom Graphics in SAS® Output.” Proceedings of SAS Global Forum 2009 Conference. April 2009.

Hunt, Stephen and Fairfield-Carter, Brian. “Zipping Right Along: Push-button SAS® Transfers via Command-line Invocation of the WinZip32 Executable.” Proceedings of SAS Global Forum 2007 Conference. April 2007.

Huntley, Scott. “Let the ODS Printer Statement Take Your Output into the Twenty-First Century.” Proceedings of the Thirty-First Annual SAS® Users Group International Conference. March 2006.

Huntley, Scott. “How to Add a Little Spice to your PDF Output.” Proceedings of SAS Global Forum 2008 Conference. March 2008.

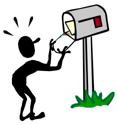
Pass, Ray and McNeill, Sandy. “PROC REPORT: Doin’ It in Style!” Proceedings of the Twenty-Eighth Annual SAS® Users Group International Conference. April 2003.

Walsh, Irina. “Pros and Cons of X command vs. SYSTASK command.” Proceedings of Western Users of SAS® Software 2009. September 2009.

Whitworth, Ryan. “Zip and Email Files Using SAS® To Reduce Errors and Make Documentation Easy.” Proceedings of SAS Global Forum 2010 Conference. April 2010.

Williams, Christianna; Hadden, Louise; Mortimore, Edward; Nagy, Frank, Plotzke, Michael; and White, Alan. “Measuring Nursing Home Quality – The Five-Star Rating System”. Proceedings of SAS Global Forum 2010 Conference. April 2010.

Zender, Cynthia. “Funny ^Stuff~ in My Code: Using ODS ESCAPECHAR.” Proceedings of SAS Global Forum 2007 Conference. April 2007.



CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Louise Hadden louise_hadden@abtassoc.com

Christianna Williams christianna_williams@abtassoc.com

Code samples are available upon request, and a zip file with a sample program and sample data is available on our sascommunity.org page: http://www.sascommunity.org/wiki/Programming_the_Provider_Previews:_Extreme_SAS_Reporting.