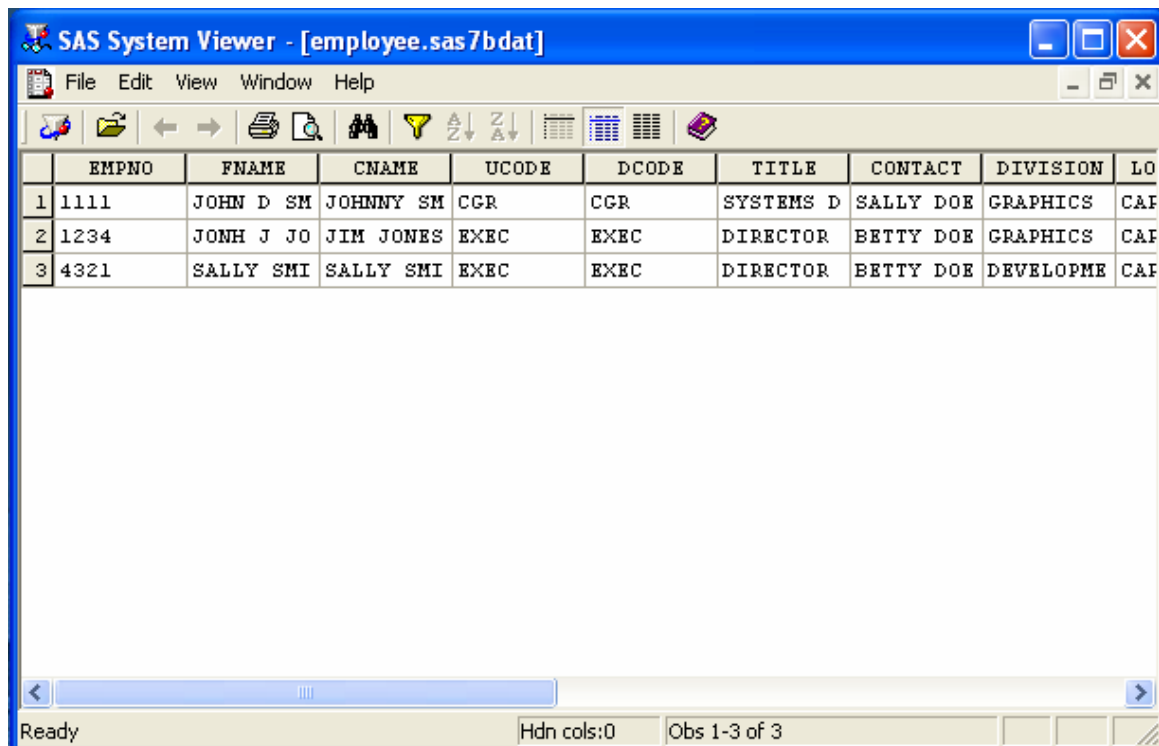


Spring 2006 OASUS Q&A

Question #1 *How can I share my SAS output with non-SAS users?*

It is fairly easy to share SAS output with people who don't have access to the SAS System. First, you can use ODS to output your results to a third-party formatted destination accessible by your target audience. Typically, you would output your results to HTML, RTF or PDF.

The freely distributed SAS System Viewer is another option to circumvent the fact that your target audience does not have a licence to use the SAS System. The SAS Viewer provides a quick and convenient way to view the contents of a SAS Data Set, Catalog, Transport, JMP®, HTML and all text based files without having the SAS System installed on your computer. You can download the SAS Viewer from the download area of the SAS customer support web site at support.sas.com.



The screenshot shows the SAS System Viewer application window titled "SAS System Viewer - [employee.sas7bdat]". The window has a menu bar (File, Edit, View, Window, Help) and a toolbar with various icons. The main area displays a table with the following data:

	EMPNO	FNAME	CNAME	UCODE	DCODE	TITLE	CONTACT	DIVISION	LO
1	1111	JOHN D SM	JOHNNY SM	CGR	CGR	SYSTEMS D	SALLY DOE	GRAPHICS	CAF
2	1234	JONH J JO	JIM JONES	EXEC	EXEC	DIRECTOR	BETTY DOE	GRAPHICS	CAF
3	4321	SALLY SMI	SALLY SMI	EXEC	EXEC	DIRECTOR	BETTY DOE	DEVELOPME	CAF

The status bar at the bottom indicates "Ready", "Hdn cols:0", and "Obs 1-3 of 3".

Question #2 After using GPLOT and GREPLAY to create many pages of multiple plots, is there a way to assemble these pages into a single file?

This question calls for another question in return: for what purpose or to what type of file? Without additional information, this question can be answered only on the basis of assumptions.

Let's begin by first backtracking a little. To obtain SAS/GRAPH output pages containing multiple plots, you begin by saving the individual GPLOT images in a graphics catalog file, and then you use GREPLAY to combine multiple plots per output page using the template feature. The result of the GREPLAY operation can again be captured in a graphics catalog file (GOUT option or statement) and, arguably, we would have all the output pages in a single file. However, this file could not be viewed conventionally and would be usable only via GREPLAY. Presumably, this is not what is desired.

There are numerous SAS/GRAPH device drivers that can send images to corresponding file types, e.g., JPEG, PNG, TIFF, GIF, BMP, etc. Many of these drivers send each image, think templated multi-plot image in this case, to a separate file. However, it is possible to create a CGM file containing multiple graphs and the pictures can then be imported into a Microsoft office document which would be both viewable and printable. The down-side is that an import filter has to be added to Office and the pictures can only be imported one at a time. A full discussion of this and other file type possibilities can be found in the SAS Technical Support document "TS-674 An Introduction to Exporting SAS/GRAPH Output to Microsoft Office, SAS Release 8.2 and higher" available at support.sas.com.

Another possibility would be to use a driver that supports PDF format. This option is discussed in another Technical Support document "TS-659 Exporting SAS/GRAPH Output to PDF Files". The PDF Universal Printer method is recommended for sending multiple graphs to a multi-page PDF file.

Question #3 My office has purchased the government suite version of SAS 9. How do I know that every feature of this software has been installed on my system?

To answer this question, it is appropriate to first paraphrase the question. Thus, because every feature of the government suite software will pertain to a licensed component, “How do I know that every component of the government suite license has been installed on my system?” To obtain the answer, you will want to compare the list of components authorized by the license to the list of components installed on your machine. This approach is applicable to any license/installation scenario, government or otherwise.

The list of licensed components can be found in the SAS Installation Data file (SID). A copy of the SID that was applied to your installation can be found under the root of your SAS installation in ...\\core\\sasinst\\sid.txt. (The default Windows installation path is c:\\Program Files\\SAS\\SAS 9.1.) Open sid.txt with Notepad or the SAS Viewer. About two dozen lines from the top, there will be a block of text under a BEGIN header (likely the second BEGIN header in the file). The first entry in this block identifies the operating environment. The remaining lines each identify a licensed component and indicate the license expiry date for that component. This is the list of components you are entitled to use.

There are numerous folders at the root of the SAS installation, some of which identify installed components. However, not all components will have distinct folders and some folder names are cryptic abbreviations of the corresponding component names. Trying to determine the installed components by looking at these folders will only reveal part of the story.

However, there is a tool available in each SAS 9.1 installation that will reveal a clear list of the installed components. This item is the SAS Installation Qualification Tool. It is available at Start > Programs > SAS > SAS 9.1 Utilities. There is accompanying documentation in SAS Qualification Tools User’s Guide.

Using the tool to determine the installed components is a simple matter. The tool requires an empty folder for output. Create a folder of your choice, for example, SASIQ at the root of your “C” drive, then launch the tool from the Start menu. A batch job window will open and you will be prompted for the folder path. Type the path and press Enter. The Window will report progress as the tool runs. It takes some time because the tool is verifying all the files of the installed components. Initial output is in the form of multiple xml files. When the verification is completed, html and pdf versions of the output will be produced. Allow this task to run until it is finished; do not click Cancel if a popup occurs.

When the tool is finished, there will be a new folder named “ivt_yyyymmdd.000” contained in the one you provided for the Qualification Tool. In that new folder, open “sasiq.htm”. In the table of contents on the left side, click “Pass/Fail results” at item

number 2 “SAS IQ Results Summary”. This table will contain a list of all the installed components. It is possible that this list will be greater than the list of licensed components, depending on how SAS was installed. Some sites might choose to install all software on the media while others might choose just the licensed components. Also, the installed NLS languages will be listed as components whereas the languages do not appear in the license.

Question #4 How to hide the content of a macro variable that contains a password?

1) PROC PWENCODE

This is a SAS Procedure that encrypts a specified password. The encrypted password can then be used in other SAS procedures or statements where a password is required. SAS will automatically decrypt the password when referenced.

Pros

- True password is not evident to users (only encrypted value seen)
- SAS will automatically decrypt as required (no special proc or step required to decrypt)

Cons

- Even though the true password is not known, the encrypted password can be used anywhere within the SAS environment (as SAS will automatically decrypt)

Example:

```
/* encode password */
filename pwfile temp;
proc pwencode in='mypass1' out=pwfile;
run;
/* store password in a macro variable */
data _null_;
  infile pwfile obs=1 length=1;
  input @;
  input @1 line $varying1024. 1;
  call symput('dbpass', substr(line,1,1));
run;
%put password="&dbpass";
```

2) Using a Stored Compiled Macro

Many of us are familiar with compiling SAS macro programs. As a default, these programs are compiled into a catalog called “SASMACR” in the SAS Work folder. However, it is possible to compile SAS macro programs to a catalog (still called “SASMACR”) but to another folder that is permanent. Thus, embedded passwords could exist within the compiled macro program.

Pros

- Compiled macros cannot be edited (source code hidden unless explicitly included)
- Easy to compile and reference

Cons

- Is possible to use software to examine contents of compiled macro catalog (i.e. Quick View Plus)
- Macro must be recompiled if password is changed

3) Compiled Data Step

A password could also be stored in compiled SAS Data Set. Once a data step has been compiled (using / PGM=) it can be retrieved and run from within SAS (using DATA PGM=);

Pros:

- Easy to compile a SAS Data Step

Cons - Is possible to use software to examine contents of compiled Data Step (i.e. Quick View Plus)

Example:

```
data _null_ / pgm=work.test;

    CALL SYMPUT('passwd','mypassword');

RUN;

data PGM=work.test; run;
%PUT &passwd;
```

4) Using a Compiled SCL program.

Like a SAS Macro program, it is possible to compile a SCL entry that has an embedded password. The compiled portion of the SCL entry is attached to its corresponding Frame entry. Thus, it is possible to delete the SCL source entry before delivering to clients. This would mean that the delivered catalog would only contain Frame entries (with compiled SCL attached).

Pros - Compiled SCL can be attached to a FRAME (Thus removing source SCL before software delivery)

Cons - Is possible to use software to examine contents of compiled SCL / FRAMES (i.e. Quick View Plus)

- SCL entry must be recompiled if password is changed

5) Custom Encryption Routine

It is also possible to write one's own encryption routine and use it combined with one of the above mentioned suggestions. This would allow great security, but at a cost; greater complexity.

Question #5 For Grid Computing: we have about 30 gigs of data. How will you do the “data split” in terms of size? Will the data transmission actually “even out” the benefits of Grid computing?

There is no one answer on how data should be split in a Grid environment. Many factors such as, number of available nodes, network speed, as well as the type of processing to be performed will determine the best scenario for Grid Computing. The way in which data is processed may also influence how it should be split. For example, if an application processes using a BY Group such as Province then a logical split might be by data for each Province. If many passes or repeated steps are required on ones data, then it may be beneficial to split data and send to available nodes. However, on the other hand simple processing may not warrant the extra time required to transmit data to available nodes. In any case the right answer will be more trial and error.

Based on various research performed by SAS Institute on large processing applications, it was clear that Grid Computing had the advantage of reducing the overall processing time of the overall task while leveraging existing hardware. The processing time reduction attributed to parallel processing may far outweigh the overhead of data transmission.

Question #6 How to read a table contained in an HTML document into a SAS dataset?

HTML is a markup language that is used for formatting document for the Web. Contrary to XML, HTML is concerned about the display of data elements and not their definition. So, it is not easy to read in a table embedded into an HTML page. For simple HTML page with only one table, one can read it into Excel and the resulting spreadsheet should be easily imported into SAS.

The other option is to use a data step and read the html page as a data file. One would need to parse the hml tags and detect the data cells of the table of interest. Not an impossible job to do, but tedious and error-prone.

Question #7 How to get the proceedings of a SUGI conference?

Although available in the past, it is no longer possible to purchase a hardcopy or CD-ROM version of SUGI proceedings from SAS Institute. An alternative to obtain past SUGI proceeding documentation is to get in touch with someone that has actually attended. Each attendee receives a complete proceedings disk in Adobe Acrobat format (in the distant past attendees received paper copy of proceedings). If a past attendee cannot be found the next best thing would be to visit SUPPORT.SAS.COM web site at the following URL:

<http://support.sas.com/events/sasglobalforum/previous/online.html>

All proceedings back to SUGI 22 are included. An advantage to using the web site is that you can use a global search that will scan all the SUGI proceedings available online.

Question #8 Can SAS/GIS provide better maps than MAPINFO or ARCGIS?

A geographic information system (GIS) is a tool for organizing and analyzing data that can be referenced spatially, that is, data that can be tied to physical locations. Many types of data have a spatial aspect, including demographics, marketing surveys, customer addresses, and epidemiological studies. A GIS helps you analyze your data in the context of location.

It is difficult to say whether or not SAS/GIS provides better maps than other GIS products such as MAPINFO or ARCGIS. Those products are full-blown commercial Geographic Information Systems (GIS) and they produce excellent maps. SAS offers two products that can deal with maps: SAS/GIS and PROC GMAP in SAS/Graph.

SAS/GIS software enables you to do more than simply view your data in its spatial context. It also enables you to interact with the data by selecting features and performing actions that are based on your selections. SAS/GIS software draws on the capabilities of SAS and enables you to access, manage, analyze, and present your data easily.

SAS/GIS software uses two basic types of data:

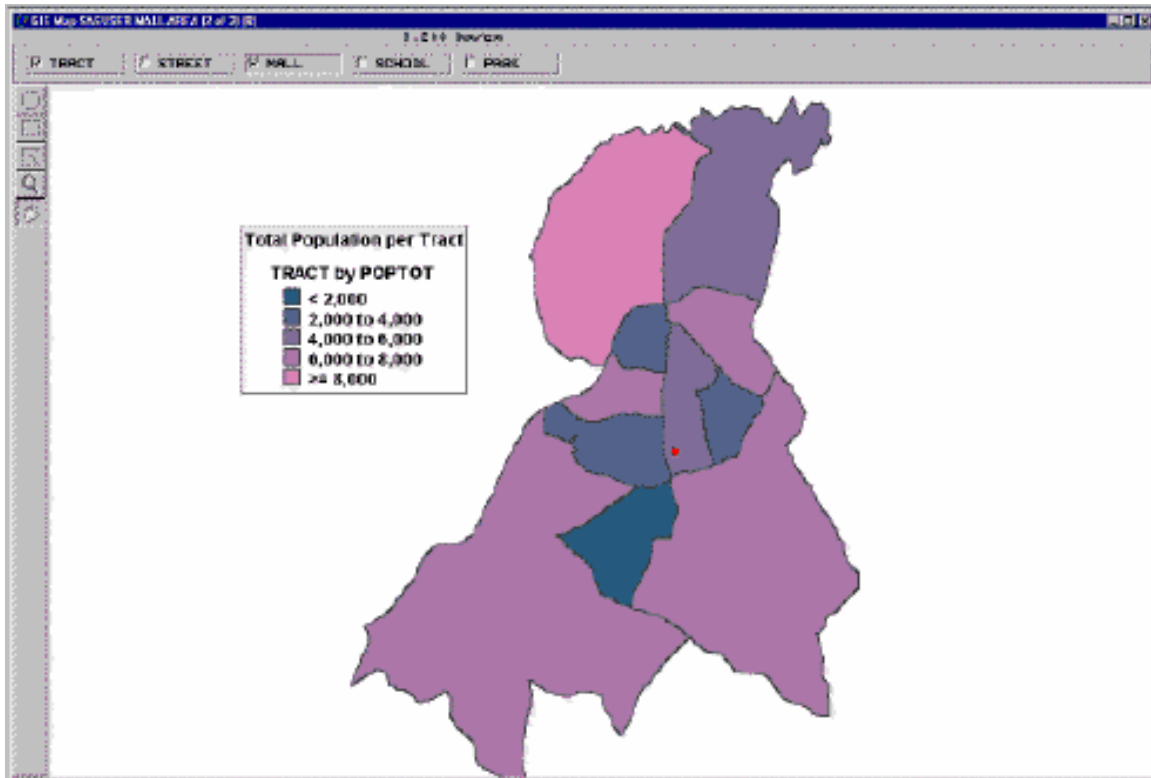
spatial data

contains the coordinates and identifying information that describes the map features such as streets, rivers, and railroads.

attribute data

is the information that you want to use for analysis or presentation. This information must be spatial in nature. Sales figures for each of your store locations, population data for each county, and total income for each household in a region are examples of information that is spatial in nature because the information applies to a specific geographic feature

A user can operate the GIS software either interactively using the GIS command or in batch using PROC GIS. The following is a screenshot of the GIS interactive environment

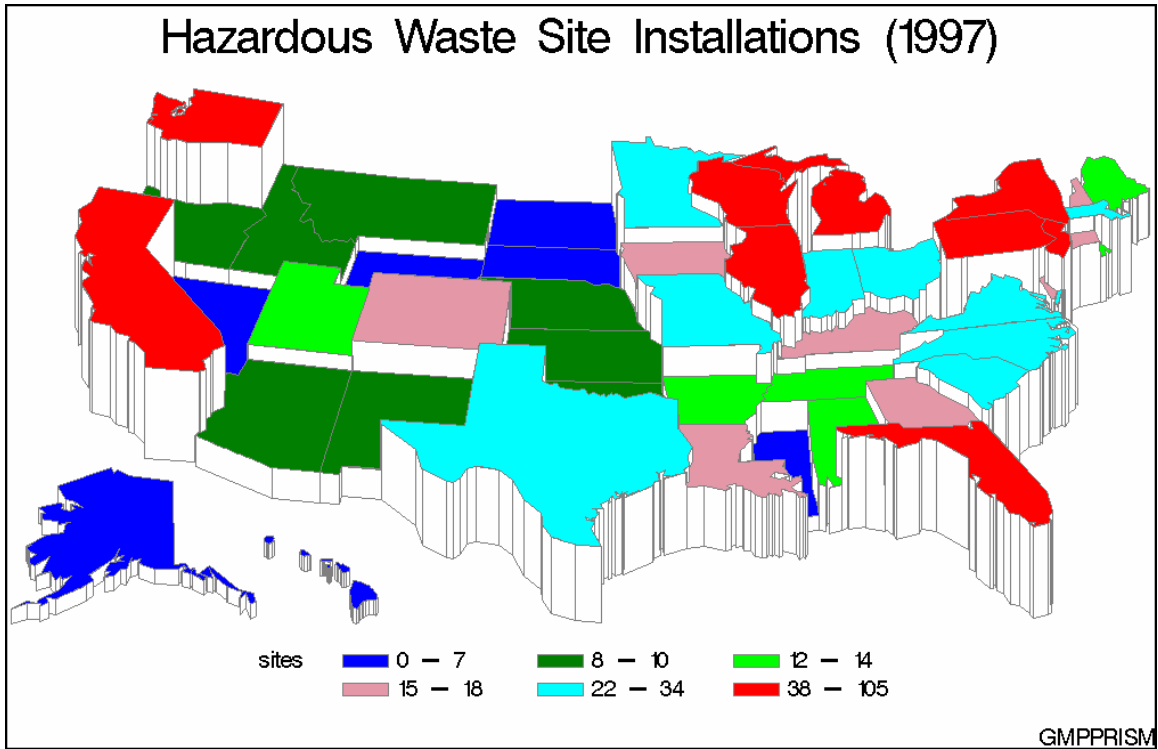


PROC GMAP is another means to deal with maps within SAS. PROC GMAP is part of SAS/GRAPH and produces two-dimensional or three-dimensional color maps that show variations of a variable value with respect to an area.

Map data sets and response data sets are used in the GMAP procedure. Depending on the type of map data set used, the map and response data sets can be used individually in the GMAP procedure or merged into a single data set to be used in the GMAP procedure. Each data set must contain the same identification variable.

There are two types of map data sets: traditional map data sets and feature tables. A traditional map data set is a SAS data set that contains coordinates that define the boundaries of map areas, such as states or counties. An alternative to using the traditional map data set is the feature table. While the traditional map data set stores the spatial information across multiple observations, the feature table uses a data arrangement to store all of the spatial information in a single variable value. The feature table's data arrangement uses the \$GEOREF SAS/GRAPH format.

Below is an example of a map produced using PROC GMAP.



The two products are similar but SAS/GIS is much more interactive whereas PROC GMAP generates excellent graphics. Enterprise Guide provides a task for PROC GMAP and also a facility to create a MAP feature table.

Question #9 How do I send attachments using Lotus Notes?

In Base SAS, you can send email messages and attachments by using the Email Access Method. Before you do so, you need to set the appropriate SAS email options because SAS needs to know what email system and what email account to use. These options must be specified in the SAS configuration file or at startup (but not using the option statement).

The EMAILSYS option identifies the email system to be used by SAS. It can be set to the following:

- MAPI (Messaging Application Program Interface) for Microsoft products such as Outlook..
- VIM (Vendor Independent Mail) for IBM products such as Notes.
- SMTP (Simple Mail Transport Protocol) provides a direct communication between SAS and a SMTP server.

So as can be seen, EMAILSYS must be set to VIM in the case of Lotus Notes. Further, it is likely that you have to specify a logonid (using the EMAILID option) and a password (using the EMAILPW option). This also applies to MAPI.

Once your email configuration is done, you can start using the email access method in your SAS program via the FILENAME and the FILE statements. Because email is more than just sending a piece of text, several options pertaining to a specific message should be set. The destination address is mandatory and can be provided using the TO keyword. You should also provide a subject using the SUBJECT keyword. To send an attachment, you should use the ATTACH keyword. In any of these options, you can provide a list enclosed in parentheses and not just a single element. The body of the message is provided by actually writing to the file using put statements in a data step.

In the example below, we use email to notify someone of the outcome of a Proc Freq. We essentially provide the recipient with the return code of the procedure and the output saved in a SAS dataset is attached to the message. Note that using a macro with conditional macro statements would allow customizing the email message depending whether or not the run was successful. Indeed, there is no need to send the dataset when the Freq procedure ended abnormally!

```

/*****
/* Example of a filename statement with the email option.  */
*****/
filename msgout email to="email@somewhere";

/* Run Proc Freq */
proc freq data=sashelp.air;
  tables date/out=freqres;
run;

/* Grab the return of Proc Freq */
%let freqrc=&syserr;
/* set location of the results to be emailed*/
%let fname=%sysfunc(pathname(work))\freqres.sas7bdat;

/* Email the results */
data _null_;
  file msgout subject="SAS Job terminated"
            attach="&fname";
  put "The SAS job has terminated.";
  put "The resulting report is attached.";
run;

```

To learn more about the emailing messages and attachments using SAS, the following resources can be used:

- SAS OnlineDoc: look at the FILENAME and the PUT Statements. Also, do not hesitate to review the email options available for SAS in general but also specifically to your platform.
- SUGI Paper 178-29: “You’ve Got Mail – E-mailing Messages and Output using SAS EMAIL Engine”, Jeanina Worden and Philip Jones.
- SUGI Paper 039-29: “Automated distribution of SAS® results”, Jacques Pagé, Les Services Conseils HARDY, Quebec, Qc
- OASUS Presentation Nov. 1st 2005 General meeting: The FILENAME Revisited.