Paper 164-27

Using Visual Basic® to Customize a Set of SAS® reports

Kevin McGowan, Analytical Sciences, Inc.

Abstract

This paper describes a software system using and SAS and Visual Basic to run a series of SAS reports that can be customized at run time. The system uses a Visual Basic front end program to collect data from the user to customize reports as they are run. The system allows the user to easily select many different parameters to be used to create a report without changing a single line of the SAS code in each report. This system reduces the likelihood of errors that come from manually editing many SAS programs every time they need to be run. The system also routes the report output files to project specific network directories so they can be stored in a standard format

Introduction

In many situations scientists have a set of standard reports they use to analyze data from many different studies. In certain cases the reports can be run more than one way and there are options that can be added to customize the reports. This paper describes a system of using SAS with a Visual Basic "front end" program that uses a graphical user interface (GUI) to allow the user to customize SAS reports at run time. The data used in this example is collected from short term toxicology rodent studies done by the National Toxicology Program to help determine if a given chemical is a carcinogen.

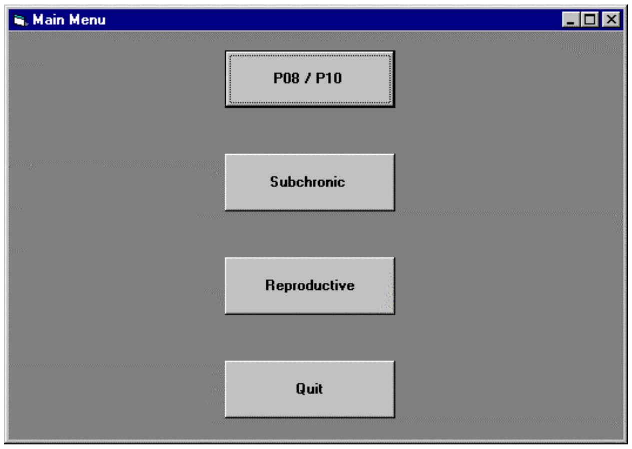
The data sets analyzed by these reports typically contains two types of variables: data that describes the animal the data was collected from (sex, species, age, body weight and food consumption) and the many different measurements that are taken during the course of the study (blood cell counts and other blood tests such as liver function tests.) A study could have as many as 50 different types of measurement variables collected and the number of animals can range from 50 to 400. There are about 20 different reports that are currently used to analyze the data New reports are developed and added to the system on a ongoing basis. The reports use a variety of statistical techniques to produce standard numeric reports as well as reports that use SAS/Graph to produce curves.

Description of the system

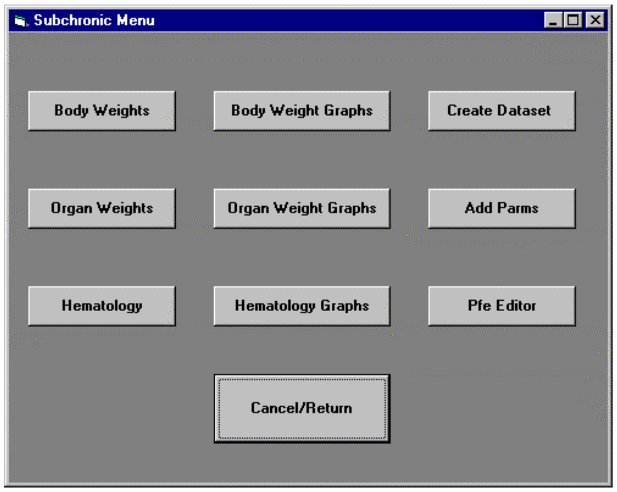
There are two features in SAS that make it possible to use a Visual Basic front end GUI program to customize SAS reports at run time: macros and include files. These features are typically used by programmers to make programming easier by reducing the number of lines of code in a program. This system uses macro variables and include files to change the input data, output format, or statistical procedures that the program uses to analyze the data. The front end Visual Basic GUI program is the method the end user uses to run the SAS reports. The user does not need to know any SAS programming, or even anything at all about SAS, to run the reports. The GUI collects all the necessary information from the user and then runs the SAS program as a batch job. The program follows Windows programming standards so anyone who has used Windows will only need minimal training to run the system. When the SAS program is finished running, the output from the report is placed into a directory that has been created for each study.

The GUI program is made up of 3 main steps:

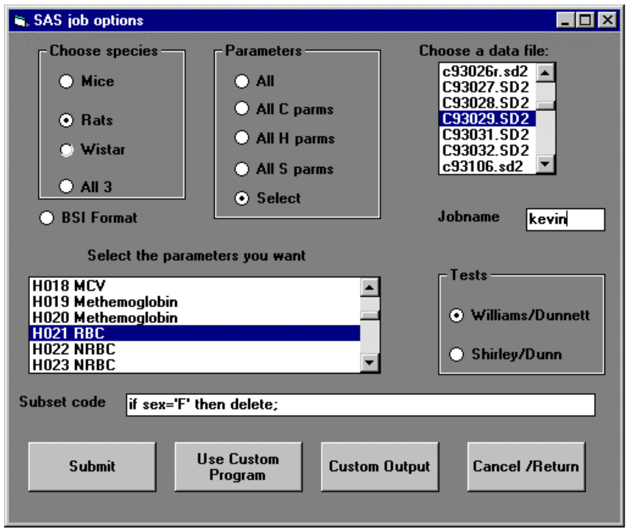
The user selects the group of the report from a provided list of options. Here, the current options are subchronic study and reproductive study:



The user selects the specific type of report. This example currently includes 6 types of reports:



The user selects the type of study, study number and the various parameters used to run the report :



Choose species: Mice [✓ Rats] Wistar All 3

BSI Format

Parameters: All / All C parms / All H parms / All S parms / [✓ Select]

Choose a data file:

```
c93026r.sd2
C93027.SD2
C93028.SD2
C93029.SD2
C93031.SD2
C93032.SD2
c93106.sd2
```

Jobname: kevin

Select the parameters you want

```
H018 MCV
H019 Methemoglobin
H020 Methemoglobin
H021 RBC
H022 NRBC
H023 NRBC
```

Tests: [✓ Williams/Dunnett] Shirley/Dunn

Subset code: if sex='F' then delete;

Once the steps listed above are completed the GUI program checks the following items before running the SAS program in batch mode:

- Does the data exist and is it in the correct format?
- Does the output directory exist so the results will be copied to the correct place?
- Is the correct output format chosen?

An important feature of running the SAS code from a GUI is that the GUI only offers the user options that are valid for the type of program they are running. For example, if the user chooses to run a subchronic program, the only parameters they are allowed to select are the ones that apply to subchronic data. Additionally if the user chooses to run a program with the data restricted to the rat species, the parameters available are further restricted to those that apply to that specific species.

Advantages of Using a GUI to Run SAS Programs

There are many advantages to using a GUI to run SAS programs. These include:

- The system runs the correct report for the type of data the user selects, and prevents the user from running reports that may not be appropriate for the data selected.

- The user does not need to edit and change any of the SAS reports to get a custom report, which eliminates the chance that the user will accidentally change a part of the code that does not need to be changed.

  In addition, since the main source code does not change, it would be possible to store the SAS source code files as read-only so that they cannot be changed by the most users. Since some of the reports contain over 1000 lines of programming code, tracking down a bug introduced by a change in source code can be difficult and time consuming. This system eliminates this problem.

- Storage of the report output files is automated by this system. Over time a large analysis project can generate a lot of files for programs, data, and reports. All of these files can end up in various directories that are not always maintained as well as they should be. The system names the output file and routes the output file to the correct directory based on the parameters chosen by the user thereby reducing clutter in the directory structure.

SAS programming techniques used

The SAS programs that are used in this system were developed and used before this system was put into production. Most of the SAS code was not changed but the input and output sections were replaced by macro variables and include files. Sections of the code that control various statistical techniques were also changed. The total number of lines that were changed amounted to less than 5% of the programs. The most common technique used in this system is macro variables. The GUI program produces an output file based on the various parameters selected by the user. The output file is then used as an include file in the SAS programs.

For example if the user decides to run a report on study 51342 using male rats and the parameter to be analyzed is red blood cell count (RBC), the output file from the GUI program will contain these lines:

```
%let isex=Male;
%let ispecies=Rats;
%let study=51342;
%let iparam=RBC;
```

Additionally these macro variables are added to the include file based on parameters chosen by the user to determine the statistical technique, computer drive, and output format:

```
%let tech=WD;
%let outfor=BSI;
%let drive=S;
```

The output file is called "includep.sas" and it is written to a temporary directory. The SAS programs include the statement:

```
%inc includep;
```

at the top of the program. One big advantage to using an include file is that after the program is run the user can go back and look at the include file to see the parameters that were chosen to run the program - the include file is not erased when the program is run. This can be valuable if the report appears to have problems – the include file can be checked to see if the parameters chosen were the cause of the problem. A planned enhancement to the system will store all the parameters for each time any program is run in a database. This will create an "audit trail" that can be examined to find out how many different ways the program was run in the past. The audit trail will also be useful in determining future enhancements to the system.

If it turns out that certain parameters are often chosen together it may be possible to group those parameters as one choice in the GUI program. Any number of macro variables or any other SAS programming statements can be written to this file or to other files that are included in the SAS program at run time. Filename statements and by group statements are also used frequently in the system. The include files can be placed anywhere in the SAS program since they work just like inserting a macro into a program. In the example above some of the macro variables are used in one part of the SAS code to subset the data to animals that meet the parameters the user selected. The SAS program contains this subsetting code:

```
data animals;
   set c&study;
   if species=”&ispecies” and sex=”&isex” and
param=”&iparam”;
```

As well as using the parameters to subset the data, the parameters chosen by the user can also be used to select portions of the code to be executed that apply to the parameters. For example the SAS code to execute a portion of the code that is only applicable to the species mice looks like this:

```
   %If &ispecies=Mice  %then  %do;

   [a series of SAS statements to be executed
only for the species mice]

  %End;
```

Dividing the SAS source code into sections that are executed based on user chosen parameters (via macro variables) makes the code more general – in this example there is no need to write one program for mice and another program for rats – the code for both species is contained in one program and the user entered parameters are used to choose which sections of code are executed each time the program is run. Another example of using a parameter to select a section of the code to be executed is this statement which uses a parameter to choose which statistical technique to be used:

```
   %If &tech=WD  %then  %do;

   [a series of SAS statements to be executed
only for Williams-Dunnett technique]

  %End;
```

Another useful technique is the use of the parameters directly in the output of the report. For example the species parameter can be written directly to the program output in the title statement by using this statement:

```
   Title “Tumor Report for &ispecies”;
```

In order to store the report output in the correct directory several parameters are combined to create the name of the output file along with a directory structure name to store the file. This example shows the use of the SAS filename statement to combine parameters:

```
  Filename prout
“&drive:\bioassay\report\output\&study\tech..txt
”;
```

As noted above, one function of the GUI program is to make sure that the output directory the SAS program will write to actually exists. If the GUI program determines that the directory selected does not exist it will not allow the user to run the program.

The GUI also allows the user to place one line of SAS code directly into the chosen SAS module at run time by simply typing the code into a text box in the GUI. Typically this code will be inserted to delete records that are to be excluded from the analysis. An example of this type of code would be:

```
   If  weight > 300 then delete;
```

Future Enhancements

Since this system is written in Visual Basic, it can only be used under the Windows operating system. Plans are being made to convert the system to Java, ASP, or some other web based software system so that it can be run by users with other operating systems and possibly by users at remote locations. Parts of the system will also be converted to use a database so that end users can change some features of the system that can now be changed only by recompiling the GUI software.

Conclusion

The system described in this paper allows end users to easily customize SAS reports at run time by selecting various parameters using a Visual Basic GUI program. The system is also designed so that the correct type of analysis is performed based on the parameters chosen by the user. The parameters are passed to the SAS program using macro variables and include files. This system lets the end user concentrate on developing better SAS code without having to spend time editing existing programs every time they need to be run with different parameters.

Acknowledgements

I would like to thank Richard Morris of NIEHS for giving me the initial idea to come up with this system. I would also like to thank Julie Scott and Laura Betz of ASI for testing and using the system and providing me with valuable feedback on ways to improve the software. Pat Crockett from ASI also provided me with valuable advice on this topic. I would like to give special thanks to my wife Lib for help with proofreading and general writing advice.

This research was supported by the National Institute of Environmental Health Sciences under contract number GS-10F-0351K.

Contact Information

Kevin McGowan
Analytical Sciences, Inc.
2605 Meridian Parkway
Durham, NC 27713
(919) 544-8500
(919) 544-7507 (fax)

kmcgowan@asciences.com