Developing a National Health and Nutrition Examination Survey (NHANES) SAS®/Web-based Quality Assurance Application

Jane Zhang, Centers for Disease Control and Prevention/National Center for Health Statistics
Edward Stammerjohn, CDC/NCHS
Tim Tilert, CDC/NCHS
Lewis Berman, CDC/NCHS
Allan Fisher, Harris Orkand Information Services

ABSTRACT
The National Health and Nutrition Examination Survey (NHANES) is a comprehensive survey which collects data on the health and nutritional status of the United States population through personal interviews and an extensive physical examination in a Mobile Examination Center (MEC). Continuous quality assurance (QA) is used to ensure high quality survey data. One aspect of QA is the data review process. To expedite data review, the QA Team has developed numerous batch SAS® programs. However, the application of these programs requires a large degree of manual effort and has not been standardized. It also has limitations with respect to reducing the time of the data release cycle.

A Web-based application was developed by integrating Sybase® with SAS/IntrNet® to automate the data review process allowing a user without SAS programming skills to perform real time data review and display tabular or graphic reports directly inside a Web browser. This application significantly improves the speed to review survey data and, by extension, helps reduce the time of the data release cycle. The specific challenges of this application include integrating the Sybase database with SAS/IntrNet, developing a user interface to query multiple survey components, and standardizing the report and error checking.

1.0 INTRODUCTION
NHANES is designed to monitor the health and nutritional status of the U.S. population. In 1999, NHANES became a continuous survey. The survey sample selected each year is a multi-staged probability sample of persons of all ages and is representative of the non-institutionalized U.S. civilian population. Data are released in two year cycles. Participation in the survey is voluntary. Findings are reported for the total U.S. population, as well as for selected race/ethnicity groups such as African Americans and Mexican Americans living in the U.S.

NHANES data are obtained by personal interviews, health examinations, and laboratory tests. All data collection methods follow standardized protocols. Initially, people who are selected for the survey samples are interviewed in their homes. The interviewed individual is then invited to participate in a health examination component which is conducted in a Mobile Examination Center (MEC). Examinees receive a preliminary report of their examination findings at the conclusion of the MEC exam and a final report of findings after all laboratory processing is completed.

After NHANES data are collected and before being released to the public, NCHS analysts review the data to assure that it is accurate, complete, and confidential. The data review process is comprised of a number of
techniques for identifying anomalous data. Historically, analysts required a variety of ad hoc reports for data analysis. Typically, programmers would develop SAS code to query the NHANES Sybase database resulting in hardcopy documents. These requests usually involved performing such base SAS functions, as PROC FREQ, PROC UNIVARIATE, MEANS, etc. In addition, more sophisticated reports were produced.

After the current survey had been in the field for about four years, it became apparent that many of the reports were similar in scope. For example, each examination component (blood pressure, oral health, vision, etc.) has an elapsed time for each respondent that is important to study for assessing workflow and health examiner performance. The same exact style of timing report can be used for each examination. Also, analysts request reports summarizing all available examinations, which can be delineated by gender, MEC location, or health examiner.

The ad hoc reporting mod of operation has the following drawbacks:

1. It is time-consuming. Requests from the analysts are scheduled and compete with other programming requirements.
2. It is paper intensive.
3. The SAS programs are hard to maintain since they are shared among several programmers. Compounding this, analysts require reports to be tweaked several times; however these changes are reflected in reports on an individual component rather than all components.
4. Many versions of the same basic program existed for slightly different purposes. Separate programs might have been written to provide similar reports for oral health and for blood pressure, or for reports by gender and by age group.

Consequently, several programmers and analysts decided to evaluate the data review process to standardize and streamline it.

2.0 DESIGN CONSIDERATIONS

Several design options were considered for standardizing and streamlining the data review process. These options include: designing a graphical user interface (GUI) with SAS/AF® to launch SAS-based report; allowing users to run reports on demand using a series of SAS macros to be designed for that purpose; allowing users to access the data through Microsoft Excel; and creating a Web-based application in SAS/IntrNet to run reports on demand. All of these options were explored and will be described below.

The first option considered is an extension to SAS named SAS/AF. With SAS/AF extensions a programmer can create a GUI to launch existing SAS programs and reports within the SAS framework. This offered several advantages such as an easy to use GUI interface, the program could be designed to return text and graphic reports, and reports could go to the screen or to paper, as needed. However, SAS/AF has several drawbacks, namely that program code needs to be installed on each user machine, there is a large software development learning curve, and it is difficult to maintain since few of our SAS programmers are familiar with SAS/AF.

Since most analysts have some basic SAS programming experience, a SAS macro solution was also considered. Conceptually a series of SAS macros would be developed to perform a number of desired analyses. This offers the advantage that the macro can be used to allow submission of several similar reports using the same macro, by simply varying the macro parameters. It reduces the need for changes
to the underlying programs by establishing a consistent set of parameters. However, it is a disadvantage since the software is deployed to each desktop, and the macros are not user-friendly. It would require analysts to have sufficient SAS programming skill to develop supporting code.

Another option explored was using a common tool such as Microsoft Excel to allow analysts access to survey data. Others have explored similar alternatives to review SAS datasets. Conceptually, it was determined that Open Database Connectivity (ODBC) could be used inside of Microsoft Excel to read data from a Sybase database with the Get External Data feature. This was coupled with an in-house developed Visual Basic Script (VBscript) for importing SAS datasets into Microsoft Excel. Since most NHANES analysts are comfortable using Microsoft Excel and Excel comes pre-loaded on NCHS desktop computers, this was considered a viable solution. To support this option, the ODBC client needs to be installed on each machine and must be configured to connect to Sybase. With this flexibility, an analyst can do review data using conventional spreadsheet software. It does have limitations with respect to the size of the dataset that can be processed, and does not offer the statistical power of SAS. Additionally, while some pre-canned reports can be prepared, the analyst would be primarily responsible for creating reports. It does offer the advantages of a familiar GUI and the analysts could produce limited results in any manner desired.

Finally, after some additional research, a fourth solution was considered using SAS/IntrNet. In this approach, SAS/IntrNet launches a pre-packaged and sophisticated group SAS reports that can access Sybase. The analyst uses a standard internet browser to access a web page. On this page, the analyst selects the data review program to execute and various options for partitioning data. These options are passed to SAS/IntrNet as parameters, after which SAS/IntrNet initiates the corresponding SAS program, written to produce the selected report. The results are then returned to the browser. Several advantages result from this facility: 1) analysts access data through a familiar internet browser; 2) SAS does not need to be installed or maintained on each client machine; 3) SAS code does not need to be deployed and maintained to each desktop; and 4) nicely formatted reports return to the browser for review and printing. This option requires a small amount of development upfront for the creation of a web interface.

These options are compared in Table 1. The Microsoft Excel option was quickly eliminated, because in practice, it is inflexible and does not offer sophisticated analytic capability. The remaining options all involved writing standardized SAS code. The basic procedure for generating SAS code is conceptually similar in each case. Parameters are coded as either SAS macro variables or as SAS Component Language (SCL) variables. SAS code is generated based on logic embedded either in a SAS macro or in a SCL program. In each case, data on Sybase is accessed either through SAS /ACCESS® for ODBC or SAS/ACCESS for Sybase. The differences however, are in the ease of use and consequently, the user interface. SAS/AF was eliminated because it posed a significant learning curve for developers. The other two approaches (SAS macros and SAS/IntrNet) are both relatively straightforward from a development perspective. For the SAS/IntrNet approach, our developers had experience in developing Web forms, making the effort manageable. Here, the deciding factor was ease-of-use, flexibility to add additional reports, and desktop software maintenance. The Web-based application has a considerably small learning curve for analysts than the SAS macro approach and new reports require minimal front-end changes. Additionally, SAS code changes do not require software re-installation at the users’ desktop. Thus, it was decided to develop the system using the SAS/IntrNet option.
2.1. Design Architecture

The architecture for this solution consists of four primary pieces: 1) the hardware server, 2) integration and configuration of software services including SAS server, SAS/IntrNet, and a web server, 3) GUI design, and 4) SAS code.

Since this application needs to support multiple simultaneous users it requires a server (computer) with sufficient resources to handle both SAS server and SAS/IntrNet. To support this effort a Dell Pentium 4 server, outfitted with 256 Megabytes of memory, 40 Gigabytes of disk space, and a 1.8 Gigahertz processor, was selected. The hard-drives are configured as a level 1 Redundant Array of Independent Disks (RAID) used to mirror hard drives for data failure protection. The server was configured with Microsoft Windows 2000 Server to handle the Web services.

With NHANES data residing in a Sybase database, ODBC was the selected method of interfacing the SAS/IntrNet software with the NHANES data. Combining SAS/IntrNet software with ODBC connectivity has the added advantage of requiring only one user account to access the database. Thus all requests to the database are generated at the server level through this single user account which greatly simplifies database account management.

The interface was developed using the Visual Basic Script (VBscript) programming language and Active Server Pages (ASP) to take advantage of the Windows 2000 Server built-in Internet Information Services (IIS) Web server. Through IIS, executable programs can be called from user entry screens. This is the method used to call SAS/IntrNet from the user interface. The main component of the SAS/IntrNet software is an executable broker program which is capable of handling calls from a Web server such as IIS. The broker program, in turn, calls any number of pre-written SAS macros containing the appropriate data review algorithms.

The user interface is where many of the advantages of the Web-based application can be seen. The first field in the user interface, as shown in Figure 1, contains the list of SAS programs available to the user. This field permits the user to select from a number of data review reports, for the examination components, on a single screen. This is a key feature of the application in that all of the available SAS programs are now centralized on the Web server. This allows the developers to deploy numerous SAS programs for a virtually unlimited number of users, all at one single location, greatly reducing software installation and technical support.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SAS/AF</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>High</td>
<td>Low</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>SAS macros</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Medium</td>
<td>High</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Microsoft Excel/ODBC</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Low</td>
<td>High</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>SAS/IntrNet</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Medium</td>
<td>Low</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 1. Summary of Design Options.
Many of the remaining 14 fields are used for class options and filtering which allow the user to both drill-down through the data and group by such characteristics as age, gender or ethnicity. By employing a standardized variable template in all of the SAS macros to accept the inputs from these fields, each of these class options and filters can then be applied to the data regardless of which program is selected. For example, by setting the Start Age to 12 and the End Age to 49, a user can filter the data to include only those survey participants aged 12 to 49 for the Response Rates program, the Mean Time program, or for any other program that is available to the user. This allows for maximum flexibility between the programs.

By design, most of the fields in the user interface are drop-down list boxes. One feature of drop down list boxes is that the user is not required to type. This has dual benefits; it speeds up the process by eliminating typed entries and it also reduces data-entry errors when making a selection. Another feature of drop down list boxes is that they can be dynamically populated with values from a database. The interface can, in effect, adapt to changes in the survey. As soon as information is added to or removed from the survey, and subsequently the database, it is automatically represented in the user interface. For example, the fields Start Stand and End Stand refer to physical locations where the survey is administered. Should the survey be administered at a new location and an additional stand gets added to the database, this new stand will automatically show up in the Start Stand and End Stand fields in the interface and can then be applied to any of the available programs. This is accomplished without any additional programming as these drop down list boxes are populated directly from the database.

Once the user selects a program and all the desired class and filtering parameters for the program, a single click on the Submit button will call the SAS/IntrNet broker program, which will then call the corresponding SAS macro. Upon completion of the SAS program, the default output from SAS/IntrNet
is Hyper-Text Markup Language (HTML), which is then displayed in the user’s browser. The user can bypass the default HTML output and send the results to a printer if desired. The default HTML output is very robust, however, and is capable of handling enormous amounts of output. This is an added efficiency for the Web-based application as users can simply browse through results in a browser window as opposed to leafing through what can amount to hundreds of sheets of paper.

3.0 PROCESS IMPROVEMENT AND RESULTS

The system presented has a number of process improvements over the batch oriented process, which had been the standard operating procedures for data review. These improvements can be grouped into four categories:

1. Interface design.
2. Standardization and improved output formats.
3. Reliance on metadata for dynamic changes.
4. Flexibility for the analyst.

A major improvement came from the development of the Web-based interface. The Active Server Page-based interface allows analysts to directly retrieve various pre-written SAS programs to perform corresponding tasks. In contrast, these tasks used to be performed under a batch process by the programmer. This required the analyst to prepare a request for the programmer; subsequently the programmer writes and debugs a SAS program and generates a report. The whole data review process typically took several days or weeks. The time is now shortened to a few minutes for each data review iteration, assuming that the pre-packaged reports cover the inquiry. Now analysts only need to select a query on the screen and specify criteria to produce the desired reports instantaneously. Furthermore, reports are generated in tabular and graphic format, which resulted in great consistency, easy comparison, and standardized graphic representation.

The new Web-based application stores only one copy of each SAS program on the server and thus enforces standardization. Care was given to select only the best SAS programs for specific tasks by combining similar programs and eliminating redundant ones. When data review was performed in batch mode, multiple versions of similar or identical SAS programs existed. Over the years, many versions of the same basic program existed for slightly different purposes. Separate programs might have been written to provide similar reports. Such proliferation of core SAS programs sometimes caused inconsistencies and inaccuracies. This presented a major challenge for programmers to modify the code. The Web-based application greatly simplifies the maintenance of SAS programs by using only one version of each program. Any modifications or changes are made directly to the single copy of the SAS program on the server. As a result, the accuracy of data review was increased with significant time savings.

The new Web-based architecture relies on an extensive set of NHANES metadata residing in a relational database. This metadata provides detailed attribute information about every single laboratory measure, question, and physical examination measure in NHANES. The metadata is used to dynamically drive the SAS code. The automation of the Web-based application eliminates the need to manually modify SAS code when changes are made to existing survey data items, when new items are added, or items are dropped. Now SAS programs automatically incorporate this information when the metadata database is modified.

As an example of the data review process, the following 10-steps are used to review all laboratory data:

1. Print and review every 25th record.
2. Check data availability by survey stands.
3. Review mean value of numeric variables.
4. Report records outside of the age and gender range.
5. Report age and gender discrepancies between lab dataset and demographic dataset.
6. Review value frequencies for all variables.
8. Report missing values.
10. Confirm all the sample persons in laboratory dataset also in demographic dataset.

Step 7 provides a good example of the advantage of the Web-based application over the batch oriented mode. In any particular laboratory test, the valid range for test results can change during the course of the study. Using the old batch oriented process, users must have basic SAS programming knowledge to manually modify the SAS programs whenever there was a change of filtering criteria, such as the limit of detection. This requires very specific and detailed modifications to the SAS code. Such requirements were often time consuming and led to errors that later took great effort to correct. The new Web-based application remedied this shortcoming of the old process by using the metadata to automatically apply this information without manually changing the code.

Another advantage of the Web-based application is better information display. The display of data can be dynamically adjusted by scrolling, thus virtually unlimited amounts of information can be viewed on the screen. In contrast, with the old batch process output formatting is very limited and step 1 would result in reports such that each record would span several unevenly formatted physical lines in the SAS output window. Figures 2 and 3 compare the output displayed in the Web interface with the old batch process for step 1.

![Figure 2. Output Displayed in the Web Interface](image)
Figure 3. Output Displayed in the SAS Output Window

Figure 4 shows a chart of completion times for audiometry examinations by selected technicians. It is dynamically generated by selecting Program, Component, Start and End Stand, Technician ID, and Class parameters.

Figure 4. Output Displayed Technician Exam Completion Time
4.0 CONCLUSION
This Web-based application automates much of the data review effort helping NHANES staff provide high quality survey data to researchers. Process improvements include removing code redundancy, minimizing ad-hoc code modification, and producing standardized reports. It also decreases the turnaround time for report production by eliminating several steps, such as preparing specification, and writing/modifying programs. In addition, the Web-based application relies on industry standard tools, commercial off the shelf equipment and software, is easy to maintain and update, and allows for a much larger user base. The first phase of this application serves as a baseline Web-based application. Future enhancements will supply additional options and flexibility to users.

5.0 REFERENCES

6.0 CONTACT INFORMATION
Jane Zhang
Centers for Disease Control and Prevention/National Center for Health Statistics
3311 Toledo Rd.
Hyattsville, MD 20782
Work phone: (301) 458-4282
Fax: (301) 458-4029
E-mail: jdz1@cdc.gov

TRADEMARK INFORMATION
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.
Sybase is a trademark of Sybase, Inc.
® indicates USA registration. Other brand and product names are registered trademarks or trademarks of their respective companies.