

Paper 240-25

LabOne's Production Reporting Environment - Post-Web

Steven Beakley, LabOne, Inc., Lenexa, KS

ABSTRACT

LabOne has recently experienced unparalleled diversification and growth of its core businesses. This fact, along with significant changes in the diversity and magnitude of informational needs of LabOne Statistical Engineering clients, both internal and external, has created a paradigm shift in how we process, present, and distribute the requested information. We will discuss the evolution of our departmental goal to provide such value-added statistical and historical information, and the process changes we implemented to make our goal a manageable, auditable reality.

With the implementation of our corporate intranet came a change in our departmental strategy for delivering all this information. The following paper presents the strategy and the steps we have taken or will take in meeting these objectives since our web enablement endeavors began.

WHAT WE LEARNED PRE-WEB

LabOne's Statistical Engineering organization has been using SAS® since 1990 to populate a data warehouse and to produce historical and statistical reports for clients of our three lines of business - risk assessment, clinical, and substance abuse testing. As the volume and complexity of the reports that we produced increased, we began turning over the production of these reports to a computer operations organization so that we could concentrate our departmental efforts on development of other reports and applications to aid our clients in decision support.

Prior to implementation of our corporate intranet, we invested a great deal of time, brainpower, and manpower in producing a dataset- and macro language-driven SCL application that was easily maintainable and highly flexible. PRODSCL builds and executes a large batch jobstream based upon the records it identifies in a master dataset called STATLIST, which contains a record or records for each report in production. (This application is described in detail in paper #123-25, "LabOne's Production Reporting Environment - Pre-Historic to Pre-Web". Appendix III of that paper replicates PRODSCL.SAS. Appendix II shows a contents of STATLIST.)

We had complete control over the timing of production job execution, the recipients receiving the report, the output form and destination, all manageable through a simple SAS dataset. While we were ready to take the next logical step and web-enable a number of our production reports and applications, we were not ready to give up the flexibility that our previous solution had afforded us. Clearly, any web-enablement solution was going to need to be a significant enhancement to PRODSCL, not a replacement for it.

GETTING WEB-ENABLED

LabOne's first attempts at getting web-enabled actually predated the introduction of SAS/IntrNet or the HTML formatting macros. A need arose to have graphics representing specimen turnaround time throughout the day in our laboratory. Using the SAS-supplied GIF driver and a complex DATA_NULL_, we were able to develop HTML pages and GIF graphics, store these on our web server, and update them through a batch job automatically submitted every two hours.

With the release of the HTML formatting macros, we were able to implement addition reports into PRODSCL, which produced static HTML pages each night summarizing daily volume for each of our three lines of business. The report replaced a manual process that included transcribing statistics from a paper report into an email and sending it to a predetermined list of individuals.

Neither of the above processes has required any modification to PRODSCL or STATLIST in order for us to web-enable them. Both programs were specifically designed to produce static web output only. However, we have identified enhancements to PRODSCL and STATLIST that will enable us to add web output to the existing output destinations of hardcopy, fax, spreadsheet, or email for any production report.

PRODSCL/STATLIST: THE SEQUEL

The first step in enhancing our production environment to accommodate web output globally is to introduce a new reporting structure that developers will need to adhere to as they code production reports. Specifically, we will no longer code output presentation within the source program. Instead, developers are required to produce a final output dataset of formatted, ready-to-display elements. PRODSCL, after modification to include HTML as a standard output device, will determine what to do with the final output dataset according to parameters identified within STATLIST. It will also determine which elements will be displayed by querying STATLIST. A production run will generate the output dataset for the given time period and append it to the historical dataset for that report.

The front-end enhancement includes the development of a SAS/IntrNet application that will dynamically produce a menu listing of the reports available via the web based on recipient. The recipient can then produce the report by executing a broker call from a web page that will display the output from the final output dataset. An automated email will alert the recipient that the report is available for viewing.

ADVANTAGES

The changes that we have outlined introduce a significant number of advantages over our present production environment. We currently have an auditable environment, but the process of auditing is a manual one. Logs must be reviewed for errors in the event of a problem. Reports that are misplaced cannot be accessed by their intended recipient and must be manually reprinted. No history of reports can be readily reproduced. Reports cannot easily be customized by recipient, i.e., some recipient's output displays sensitive data while other's do not. Multiple runs for different recipients are made, thereby undermining the efficiency of our current process. All of these disadvantages are eliminated by our proposed enhancements:

Auditing – By producing a final output dataset, we can easily determine if a job has or has not produced any data. In the past, reports that did not produce data would still produce output and data capture problems would not be brought to our attention until the recipient received his or her report. (Because of the large number of reports we have running, to eliminate the need to restart processing in the event of an abort in a single job, we utilize the SAS option NO\$SYNTAXCHECK. This will allow processing to continue in the event of an earlier abort.) Under the proposed scenario, we will conditionally generate the output based upon values of return codes.

Misplaced reports – By accessing their own personal web “page”, a recipient will be able to reproduce any report. The Statistical Engineering department and Operations are relieved of the responsibility to find or reproduce missing reports.

Report history – Because a historical output dataset is created, the recipient can view reports back in history through their web interface by selecting the time period they want to view and reproducing the appropriate output. (See Appendix for sample report selection screen.) Processing for a particular report only need run at the most detailed time period necessary, i.e., quarterly runs are no longer necessary if monthly runs occur, because the broker call can summarize from a monthly to a quarterly level.

Customization – Reports can be customized, allowing certain sensitive fields to be displayed only for intended recipients (elements are identified within the STATLIST record for that report/recipient combination).

Multiple runs – Only one run of the program will be executed, and all combinations of companies, tests, states, facilities, and so forth, will be processed and added to the output dataset. The burden of producing individual reports is then placed on the web server delivering the reports.

CONCLUSION

The intent of the preceding paper has been to describe how LabOne’s Statistical Engineering Department has handled the growing need for information by its internal and external customers through a web browser interface while keeping the administrative overhead involved to a minimum. Our basic methodology has been presented, along with some ideas for implementation and enhancement. While some mechanisms specific to our systems and processes exist, a similar approach could certainly be taken by other organizations trying to accomplish the same thing.

CONTACT INFORMATION

If you have any questions, comments or feedback, please do not hesitate to contact the author at:

Steven Beakley
LabOne, Inc.
10101 Renner Boulevard
Lenexa, Kansas 66219
Work Phone: (913) 577-1380
Fax: (913) 888-4160
Email: steve.beakley@labone.com
Web: <http://www.labone.com>