Paper 214-30

Are You Ready to Roll? SAS®9 Implementation Lessons Learned

Tim Walters, PMP, ThotWave Technologies LLC., Cary, NC

ABSTRACT

No doubt, over the past year, you've been eyeing that shiny new SAS® 9 Business Intelligence Architecture and kicking its tires. You're probably learning more about its robust, enterprise architecture design, highly tuned engine for parallel processing, great handling around data management bumps and curves, as well as the new management console that allows your pit crew to keep the architecture in the never-ending race to provide excellent decision support. You also may have noticed the new reporting dashboard that greatly enhances your users' driving experience. If you are in the initiation or planning phase of your SAS® 9 ETL Server installation, we can offer some guidance that may be of use.

INTRODUCTION

This paper, authored by the project's consulting team lead, will present experiences from both the functional and technical dimensions for a new SAS® Architecture installation project. Valuable lessons learned will be discussed, focusing on the initial assessment, planning, requirements, design, implementation, testing, rollout and subsequent success of SAS® 9 ETL Server. Practical tips as-well-as insights to the business impact of this project to the organization will be offered that will assist you in your deployment of this revolutionary architecture.

OVERVIEW OF THE BUSINESS NEED

The company represented in this paper is in the life sciences industry, providing health insurance coverage and managed care services to their unique membership. They have grown very rapidly due to their successful business model; however, this rapid growth has increased the need for more robust information architecture to meet ongoing regulatory requirements as well as tactical and strategic decision support.

CURRENT STATE OF DECISION SUPPORT AND INFORMATICS

As a result of an internal assessment of the current state of the company's decision support and informatics function, several findings highlighted the urgent need to initiate a project to consolidate and reengineer the decision support processes, technology, and organization. The project sponsor conducted a series of interviews across the company and discovered these key observations:

- There was little or no coordination between the disparate groups that currently service the information needs of the company
- There was a clear duplication of effort in the servicing of information requests
- The competition for decision support and IT resources was not prioritized and sustainable
- The analysis and conclusions were not consistent from group to group
- There was a lack of focus on strategic objectives due to the heavy servicing of operational and regulatory information needs
- The data was not structured in a form that enhanced the decision support life-cycle

The bottom line was, the current state of the decision support process scratched the surface of the company's needs, and arguably, exacerbated the problems management were confronted with when trying to make sense of the data. The end results were that all energy was focused on the regulatory reporting demands to the exclusion of all other needs. The amount of time it took to service an information request was prohibitively long, caused managers to distrust the data, and locked the analysts in a report writing role, not a true analytical role. The chaotic nature of the decision support process spawned grass-roots efforts to fill the void, which resulted in narrowly focused, department level reporting methods that were not scalable or supportable to the enterprise level. And even more fundamental, there was no common, agreed-upon dictionary that contained definitions of metrics, measures, calculations, or locations of appropriate data elements, and blurred lines of authority in regard to the source data.

BUSINESS CHALLENGES

The project sponsor was hired specifically to sort out and solve the above deficiencies and provide a foundation to address more strategic challenges that the company faces over the next few years. The executive team had mapped out a very aggressive growth strategy, perhaps doubling or tripling the amount of covered members over the next few years. Without a firm decision support foundation in place, the company would be very hard pressed to grow at that rate, let alone manage its existing business prudently.

The executive sponsor created a two year vision to guide the growth and capability of his brand-new decision support and informatics group to meet the huge challenges from several perspectives: care management, quality of care, business management, and regulatory compliance. He set expectations with the executive team that included a much focused effort on building a strong platform for decision support in the short-term.

SELECTING THE PRODUCT

The project sponsor selected SAS® based on an objective analysis of internal technology and the business intelligence marketplace. As a long-time SAS® user, the sponsor was very familiar with the power of SAS® from a data management and analytic perspective, and very favorable assessments of SAS® by industry research groups didn't hurt either. The heavy use of SAS® by most major health care organizations was cited as a strong endorsement of the solution. The executive sponsor's vision of the future information architecture mapped point – to – point with the SAS® Intelligence Value Chain. The end-to-end data management coupled with the capabilities for high end analytics and enterprise reporting in one integrated solution was a very strong argument to the Information Technology decision makers as well.

SHORT AND LONG TERM PLANS

The executive sponsor's vision included a staged approach to deploying the capabilities of the new SAS® 9 architecture. The breadth and depth of the SAS® BI solution would be too much for the Decision Support and Analytics Group to handle in the short term, given the business priorities.

SHORT TERM PLANS

The immediate goal was to create a "modified snowflake" data architecture design that will allow the granularity of the data to remain at the lowest level. This would allow the decision support analysts to get at the data more easily in a form that facilitates advanced analytics and reporting. All the heavy lifting and complex transformations would be done by the ETL process, giving the analysts more time to explore the data and ask questions. The scope for the data structure was limited to a conceptual design of 5 major tables and a few ancillary reference tables, mainly code and description sets of procedures, diagnosis, and demographics. The structure would be supporting analytical reporting; therefore it would be refreshed every month. The schedule was laid out to deliver a base set of reporting tables, ETL processes, a core set of users and permissions, data verification reports, and a small set of analytic reports within a nine week time frame. This would give the analysts the basic set of data in order to address the more immediate, tactical decision support needs.

LONGER TERM PLANS

The future enhancements to the architecture that were out of scope for the initial effort included the following:

- Evaluation and analysis of contract performance
- Migration of legacy reports to SAS® stored processes
- Predictive analytics and data mining
- Fraud and abuse detection/prevention
- Maturation and differentiation of the internal programs for competitive advantage
- Geographic analysis – usage patterns and spatial analytics
- Benchmarking and performance management score carding
- Disease management and control
- Clinical best practices
- Enable business growth - acquisitions and management contracts

TECHNICAL ENVIRONMENT

The current technical landscape at the Company can be summarized in two words: Microsoft (MS) and Hewlett-Packard (HP). The IT strategy includes a rigorous adherence to low overhead and reduced total cost of ownership which contributes to the Company's extremely competitive posture compared to other health management companies. Leveraging one set of foundation technology simplifies discussion of the architectural framework for the decision support application as well. The standard database technology is MS SQL Server. The standard web server is IIS. The data tier resides on a Storage Area Network (SAN). There is a new SAN that has capacity for the data requirements of the SAS® solution. There is fiber channel connectivity between servers. The application databases in SQL Server, however, are currently at maximum capacity. Hewlett Packard hardware is the standard for the Company.

It is useful to note that the original data center physical plant was at its maximum capacity for physical space. An internal project was underway to cohabitate the data center to a larger area that has room for growth. This move was planned to be complete prior to the project's kickoff date. The Company had five (5) PC SAS® licenses.

IMPLEMENTATION RETROSPECTIVE

As the abstract states above, the purpose of this paper is to recount an install of SAS® 9 and offer useful take-a-ways that can be considered and incorporated in subsequent projects. To set the stage for this retrospective, I would like to quote from Kerth's Prime Directive: *Regardless of what we discover, we must understand and truly believe that everyone did the best job he or she could, given what was known at the time, his or her skills and abilities, the resources available, and the situation at hand.*[1] As long as this attitude is reflected, the retrospective can be a productive learning experience. To organize the lessons learned, I have broken down the topics for discussion into nine major areas: Project management, design, software install, security, development, ETL, metadata management, and testing.

PROJECT MANAGEMENT

The initial architecture workshop to scope the project was accomplished in two days. This was a good high level effort to define the project scope; however, the project should have been re-scoped after the detail design. For example, the scope was set at a time when the full definition of the data elements and target tables was incomplete. The scope should have been progressively elaborated as more was learned during the project execution phase. The original estimate was based on a set of 5 main warehouse tables and an undefined set of reference tables for code sets. The resulting number of main tables was thirteen (13) and the reference tables ended up to be twenty eight (28). Bottom line for project management, plan for change. The team realized early that the level of the data architecture design would be more intense than originally thought; however, we decided to continue targeting the original schedule agreed upon in the proposal. In hind-sight, the team agreed that implementing change control early to account for the increased level of design work would have been more beneficial.

The project had a month-long delay due to the move by the data center. The SAS® server could not be setup until the new data center was ready, which delayed the critical path task of installing the software. This squeezed the time available to complete the project by the executive sponsor's desired time frame. This issue was out of the project team's control.

DESIGN

The team's data analysts spent a great deal of effort to define the target data structures that were required for reporting as well as the right source data and the transformation rules to move the data from source to target. This situation was due to the fact that the company did not have adequate documentation or experience with the source data and there was no consensus on the correct business rules to use when defining measures, transformations, and calculations. This expanded the design effort considerably. There were severe limits on the availability of IT and functional subject matter experts for gathering information about the source data, and, given the lack of institutional knowledge on the right data sources, there should have been more time built into the plan to accommodate for intense data architecture design and validation (of the data to be included in the warehouse, its sources and mappings to the targets). Lesson learned is to follow prudent change management where it makes perfect sense… After all, a plan is just a model and forecast of reality. If the model is not accurate, it should be modified based on current knowledge of the environment.

The conceptual, logical, and physical design of the ETL structure (source, raw, staging, and warehouse) worked very well. ETL studio was extremely flexible and supported the complex nature of the ETL process. The up-front design of the ETL process was directly responsible for the smoothness of the development effort and the lack of major changes and rework.

The source system for the SAS® warehouse was decided to be a legacy reporting warehouse that was in the process of being retired. This was done with the understanding that the legacy source system would not be replaced in the near-term, as the enterprise could not wait for the new claims system to go on line. Data quality was a serious concern, and a parallel effort to validate the legacy warehouse was commenced. This validation effort was critical to prove the integrity of the resulting warehouse and to dispel the chronic organizational concerns about "data integrity".

Late in the deployment process, it was discovered that all date fields should be transformed from date/time to date only. The client team members assumed that this would be done, given the fact that time was not relevant in their data. The consulting team assumed that keeping time granularity was fine, given past projects and practices. This highlights the necessity to have more detailed design validation of all target tables, columns, formats, lengths, and business descriptions signed off and frozen before major development starts. This particular modification did not cause major re-work. Another late decision to pair business keys to all surrogate keys on the fact tables was made to support usability. This change did not require a huge amount of rework, but again highlights the need for interim validation of designs and tight communication.

All true reference data was normalized into lookup tables and SAS® formats were created for ease of reporting. Several data elements were de-normalized in the warehouse as the warehouse structure was not meant to be a classic, historical perspective warehouse, but the intent was for it to support analytical tasks.

The client analysts built the foundation for the table designs using an excel sheet that facilitated capturing the required source data information. This format was easy for the client team to manage and collaborate around. The sheet was then used by the consulting team as input to the ETL design process. The design document versions were tightly controlled using a web-centric document management system. Team members had the ability to subscribe to design documents and were automatically notified when a new version was posted. This greatly enhanced the speed of the workflow, while ensuring that design document versions were tightly controlled.

SOFTWARE INSTALL

The software install process for an enterprise implementation of SAS® 9 is not trivial. You no longer have the convenience of merely mounting a CD and running the setup.exe. Each install is unique to the technical platform, mix of SAS® software that is licensed, as well as the unique nature of your IT environment and user base. Don't underestimate the level of planning that needs to take place and don't take anything for granted during your planning. There is no such thing as a dumb question, as that very question may indicate an issue that could potentially be a show-stopper for the install. A question like, "Did you receive your Software Media and SID files?" may be seem like a slam dunk to those familiar with the install process, but you would be surprised what can and can't happen…

The pre-install is mainly a project management task, planning and coordinating the install of the software. This is also when you determine the hardware configuration, procurement (if it is a new box), and specifications (server model number, server CPU, server memory, storage configuration). The response to your inquiry about the hardware shouldn't be "we'll put it on some Windows box".

As in sports, its not over till its over. With this installation, it doesn't start until the hardware is ready. Along with getting the detailed specs, you need to continually ask, "What is the status of the hardware?" It is recommended that you have a resource that has been trained to install SAS® Enterprise Server software to do your environment. The process is very complex and requires a thorough understanding of the process. It is highly recommended that you have the installation instructions and any install course notes available to you during the install.

The Company had several binders of software that were sent to them because whenever a change occurred to their license agreement, a full set of media was shipped out. Also, fresh new binders were sent when version 9.1.3 was requested for the install (9.1.3 was just released prior to the install). In fact, SAS® sent 7 shipments of media within six weeks which confused the client. The plethora of software media was risky; as the correct version and CDs required us to perform a very close examination of the binders to make sure the right order number was installed to the server. The consultants were inadvertently given the incorrect set of CDs initially. Lesson: you must closely manage your software media if you find yourself with many shipments as in this example.

You may not get access to the server room, so you should use a remote desktop and mapped drive to build the software depot. We recommend that access to the internet is available to the SAS® application server to facilitate downloading hot fixes and any third party software such as Adobe 7.0 PDF reader to view SAS® documentation. In addition to the SAS® system accounts that are covered in the pre-installation checklist, you must set up admin and service accounts for the LSF Job Scheduler (i.e. lsfadmin and lsfservice). These can be domain accounts or local to the machine. It is recommended that you do the 3rd party installs first and lay SAS® on top of them. The only piece we encountered that needed this is the JDK. It does seem funny in the middle of whole install to go out to java.sun.com and download a JDK.

The actual install process should go very smooth once you have followed the necessary pre-installation checklist and set up the required system accounts. Oh, and one thing about these accounts… we highly recommend that you follow-up behind your system administrator and double-check that the setup of the system accounts were done correctly and that they match the pre-installation checklist. For example, if the sastrust and sassrv user accounts are not set to "password never expires", then when they do expire, the SAS® services will not run. This is because the sastrust and sassrv user accounts are system accounts and their passwords are encrypted and saved in about a dozen configuration files. If the passwords expire, you most likely will have to re-run the configuration wizard, which isn't something that you want to do. This issue ultimately caused a month-long delay in implementing the system. The actual software install went very smoothly, owing to the fact that our installation resource had the appropriate training and followed the installation instructions precisely.

The post installation process is mostly automated, but there are a few manual steps required to lock down the permissions to the metadata server. One gap in the documentation that we noticed is that there was no explicit step for setting up a backup job for the metadata. Eventually, the system administrator ensured that the application server was being backed up, but during early development, we were motivated by paranoia to set up a scheduled metadata backup script which utilized the %omabakup macro that we ran every two hours during development. You don't want to lose any of your metadata because you didn't back it up. We scheduled two days for installation and testing, but, it would have been better had more effort been planned in the post-installation testing process. Test scripts should have been prepared in advance to test all of the software configuration and functionality.

In regard to the LSF Job Scheduler install, one of the areas that seemed very helpful in the multi-machine install course notes is the information on the LSF/Job Scheduler. There are some pre-/post- install steps that go with this tool, and they are not well integrated with the configuration wizard (since LSF is a 3rd party product). The extra steps are covered in the LSF installation doc, but the multi-machine course has an entire section devoted to the steps that are required (Sec 8, 16 pages). Note also that LSF ideally requires 2 additional users, lsfadmin and lsfservices which were mentioned above. LSF does require an extra license that you have to request via the SAS® web site, which gets emailed to you in about 10 minutes. Seems like a good thing to get out of the way early.

SECURITY

One of the most complex aspects of the new SAS® 9 architecture is the security model. This model is extremely flexible and robust, which allows for multiple levels of security across your intelligence chain. Another driver of the complexity is the layering of host authentication on top of the SAS® security model. We recommend that all administrators of the metadata be intimately familiar with the security model and proper planning and testing be done to ensure that you get the results you expect. Most issues we encountered were due to permissions.

DEVELOPMENT

Most development was done using the very superb SAS® tool, ETL Studio. The remaining base SAS® development was done using SAS® Enterprise Guide and the good old SAS® programming client. Enterprise Guide was a splendid development tool due to the easy access to the metadata server and libraries defined therein. However, all libraries accessible through EG are read only, but, there is a work-around to allowing write access to base SAS® libnames through EG. All libnames are read only via EG unless this special configuration is done. The ETL programmers used the project repositories and change management while developing in ETL Studio. This process worked very well; however, there were times when a developer checked out an object from the foundation repository which would also automatically check out all objects associated with it. This caused other objects that may not be modified to be unavailable to other developers.

ETL

For those that are familiar with SAS® Data Warehouse Administrator, you will find that ETL studio is a vast improvement in terms of usability and functionality. The development team was very impressed with how quickly we could deploy our ETL processes using the point and click interface. Mundane tasks were very fast and easy to accomplish (i.e. Mapping, joining multiple tables, creating metadata definitions, and deploying jobs).

The transformations that come “out-of-the-box” are very useful in specific circumstances. We did have to create our own surrogate key generator transform in order to get the specific behavior that we needed. Our recommendation is that you review the underlying code that is generated from any transformations that you plan to use. These transformations are extremely helpful, but don’t assume that because it is a black box it will perform exactly as you expect. We also found that the fact table lookup transformation had some limitations as well. The generated transform code returns the business key value into the surrogate key field if the lookup table surrogate key does not exist. We created our own fact table lookup transform that handles missing dimension records more cleanly (i.e. inserting a dummy surrogate key instead of the business key). Also, each lookup becomes a SAS® format, so this can get very inefficient if you have an array of columns that lookup the same dimension table (i.e. Diagnosis – primary, secondary, and tertiary) and millions of records to process.

In terms of developing ETL processes, we found that breaking up complex transforms was easier to maintain and test. In one ETL stream, we created four separate transformation steps to split up the complexity of the process, instead of creating one large process. This approach allowed us to develop in a step-restartable fashion. We were able to unit test each step, saving off the output of each step, thereby improving the speed of development down the data management chain. The step restart-ability combined with filters and pre-defined test cases enabled us to nail down complex transformations very rapidly. Oh yes, and don’t label the job processes with spaces in the name. The LSF Scheduler doesn’t like spaces in the names of deployed jobs.

METADATA MANAGEMENT

The Management Console client was extremely helpful in terms of centralizing the management of the environment’s metadata. This is truly the most innovative and most powerful aspect of the SAS® 9 Architecture. By centralizing all metadata management tasks into one hub, SAS® is now manageable from a single point of control.

We did learn a few tricks that can save time down the road. If you are accessing a DBMS via a SAS® Libname that is created in Management Console, you should have the DBA set up a proxy account that has rights to the DBMS schema that you need to access. This proxy account should have a password that does not expire, as it will serve as the trusted access to the DBMS. Also, be advised that you cannot use the sasadm user account to create libnames for your DBMS. The libname statement includes an encrypted password when it is executed, but the sasadm user account’s password does not use the encryption algorithm correctly, for a reason. The sasadm account is for management of metadata only, not executing libnames.

TESTING

The client and consultants should work together to develop the acceptance test plan early. This is preferable, because defining a test plan up-front is an excellent guide for developers… they know what to code for. Reports and queries of record as well as process qualification tests should be done on all components of the software to ensure that it is supporting the business processes as expected.

This goes without saying that adequate time should be budgeted for the testing activity. A thorough testing plan should include interim data validation steps, not validation at the close of the project. In this project, we did not have a clear definition of the tie-out process that would validate the legacy system to the new SAS® warehouse until the very last week of the project. Our testing consisted of validation reports at an aggregate level, comparing record counts and total dollars. The planets must have aligned during this week, because our reports from the SAS® warehouse tied to reports that were run in parallel off of the legacy warehouse, with little time to spare.

CONCLUSION

This project is useful to us as a learning tool from several viewpoints. The project management view clearly uncovered several opportunities for improvement. All projects, particularly business intelligence, research, and other technology projects, require an approach that embraces and plans for change. The nature of a SAS® 9 project certainly requires a very agile approach to project management. Our initial scope was defined at a very high level and our budget wasn't flexible to allow for progressive elaboration during project execution. This project was a time-boxed schedule, which meant that there was little if any wiggle room if things did not go as expected. The goal was to get a validated warehouse up and running in as short of a time-frame as possible so that the decision support analysts could explore the data, service on-going tactical reporting requirements, uncover data quality issues, and start planning for more strategic exploitation in the near future. The goals were achieved due to several factors, some human, some process, and some technical. We highlighted many take-a-ways throughout this paper which should offer you some valuable input as you contemplate your own SAS® 9 project.

If you haven't concluded by now that the SAS® 9 Intelligence Architecture is very powerful and complex, you really should spend some time reviewing the appropriate background material, specifically, the SAS® Intelligence Platform Planning and Administration Guide. Another excellent source of information is the Install Center on the SAS® Website. These resources contain incredible amounts of information to assist you in the planning process. Planning and preparation are the two words you should take away from this paper, as they will pay off exponentially during the project execution phase of implementing SAS® 9. With good planning, training, and assistance from SAS® technical support, you should be able to quickly implement a foundation SAS® 9 architecture that will deliver immediate business results.

REFERENCES

1. Kerth, M., Project Retrospectives – A Handbook for Team Reviews, Dorset House Publishing, 2001

ACKNOWLEDGMENTS

The Authors would like to acknowledge the following project team members that contributed feedback and insight to this paper:

Danny Grasse, ThotWave
Richard Phillips, ThotWave

RECOMMENDED READING

SAS® 9.1.3 Foundation Hands-on Training for Multiple Machines Course Notes
SAS® 9.1.3 Intelligence Platform – Planning and Administration Guide

CONTACT INFORMATION

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

Tim Walters
ThotWave Technologies, LLC.
2054 Kildaire Farm Rd. #322
Cary, NC 27511
Work Phone: (919) 451-2401
Fax: 1 (800) 584-2819
Email: twalters@thotwave.com
Web: www.thotwave.com