Paper 043-2010

Lessons Learned from a BI Installation
Faron Kincheloe, Baylor University, Waco, TX

ABSTRACT

After implementing and maintaining a data warehouse for over 6 years, Baylor University installed SAS® Enterprise BI Server and migrated all of its warehouse content to the Business Intelligence (BI) platform. This paper will discuss the process that was used for the migration and lessons learned along the way. Among the topics covered will be project management, security, user acceptance, training and backups.

INTRODUCTION

Baylor University rolled out its first data warehouse content in 1999. After an unsatisfactory attempt to use SAS® EIS and Web EIS a decision was made that the warehouse would be delivered to the users in a manner that required no software to be installed on client computers. The original Information Delivery Portal did not have the security features we needed for our data warehouse. However, there was an unofficial product called the Public Sector Portal that was developed for another customer with similar needs. Along with SAS® Data Warehouse Administrator, this portal provided a sufficient platform upon which to build our data warehouse.

This web based interface to the data warehouse used MDDB Report Viewer to deliver tabular reports with drill down and reach through capabilities. Users also were provided with web forms from which they could select certain options that were passed on to Application Dispatcher programs in SAS/IntrNet® which would return standardized reports based on the user's selections. Static documents in HTML, PDF, RTF and Excel formats were added as users sought a convenient location to access reports they had previously received on paper or through email. A number of these documents were refreshed periodically by scheduled jobs that ran automatically on the server.

The Graduate School was the original "sponsor" of the data warehouse so the warehouse started off supporting primarily academic departments. Data were extracted from the student information system. The warehouse grew to support undergraduate admissions and later, information was added from the University financial system to support budget planning across campus.

In addition to the data warehouse, there was an installation of SAS® HR Vision to support human resource activities across campus. Our version of HR Vision had been heavily customized to accommodate time periods and organizational structures that are somewhat unique to a university environment. Users also accessed this system with a web browser but it contained its own independent set of user accounts and passwords.

There was also a series of reports developed in SAS® AF for our Internal Audit department. These reports ran against a combination of archived data and live data from various systems. The application itself was stored in a shared folder on the server. However, each auditor was required to have a local installation of SAS® on his or her computer in order to run the application.

All of the applications described above were housed on a single Windows server. Baylor had been archiving historical data for several years using SAS® so the warehouse data was stored in SAS® datasets. SAS® Data Warehouse Administrator was used to manage the data warehouse. The diagram below provides an illustration of this environment.
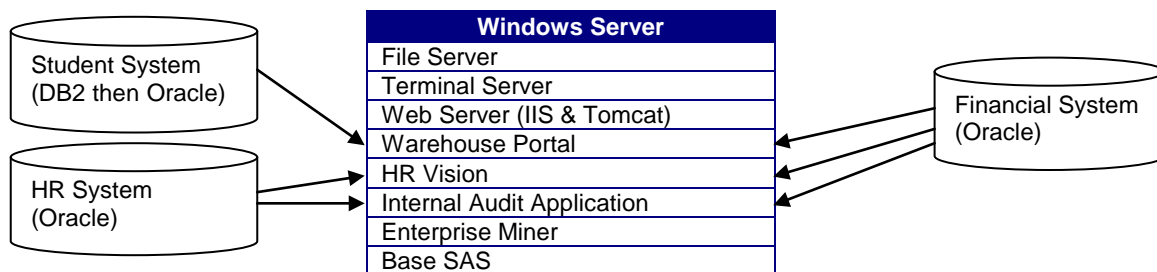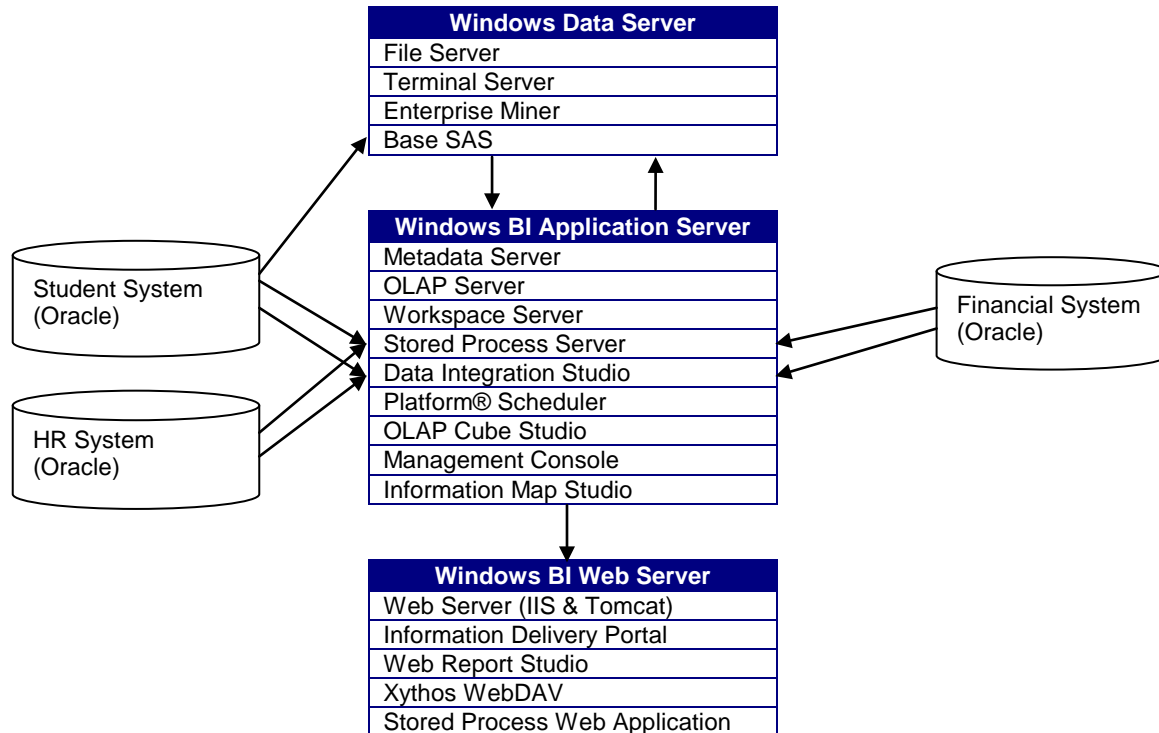

After about five years, the content structure and number of users had exceeded the security management capabilities of the Public Sector Portal. This portal was also no longer being supported by SAS. It was only a matter

of time until it would become obsolete due to changes in technology. The decision was made to install SAS® Enterprise BI Server to manage the data warehouse and make it available to campus users. All SAS® products would be upgraded from version 8 to version 9 along with this installation. Since the HR Vision product had been heavily modified, it was going to require significant modifications to upgrade it to the new version. HR Vision was being used by a relatively small number of people and a limited amount of its functionality was being utilized. This made it economically unfeasible to continue using the HR Vision product. However, a number of users were dependent upon some of the HR Vision content so this information would be incorporated into the new BI installation. The Internal Audit Application was a perfect candidate for conversion to stored processes. Since this would eliminate the need for local installations of SAS®, the Internal Audit Application would be incorporated into BI as well.

There was no budget to pay for consultants to support the BI installation in any way. The migration would be performed solely by existing staff members in Baylor's Office of Institutional Research and Testing. A three member "technology" team was charged with this task. The team leader had been using SAS® software for over 8 years prior to this project. He had previously worked in network and system administration and was the system administrator for the existing data warehouse environment. The second member also had over 8 years of SAS® experience and was primarily responsible for the manual load processes for the data warehouse. She had previously worked in Baylor's Records Office and brought with her extensive knowledge of the student data. The third member was hired to fill a vacant position in the office about the same time that the BI project began. His background was primarily systems administration. Additionally, the Director of the office wrote all of the programs to extract the data and produce the reports that had previously been provided through the HR Vision software. The technology team did not have the luxury of working full time on the BI project but the other staff members took on as much of the regular office work as possible to free up the team members to work on the migration.

Two new Windows servers were purchased to house the BI environment. One houses the web components. The other houses all of the other BI components. The server that was used for the original data warehouse was retained as a data server since a large amount of departmental information is stored there and our SAS® work is done on a terminal server from remote desktop sessions. Apache Tomcat was chosen as the web application server software because there was no cost for licensing and we had experience with this software from the original data warehouse. The diagram below provides an illustration of the new BI environment.

Windows Data Server
File Server
Terminal Server
Enterprise Miner
Base SAS

Windows BI Application Server
Metadata Server
OLAP Server
Workspace Server
Stored Process Server
Data Integration Studio
Platform® Scheduler
OLAP Cube Studio
Management Console
Information Map Studio

Student System (Oracle)

HR System (Oracle)

Financial System (Oracle)

Windows BI Web Server
Web Server (IIS & Tomcat)
Information Delivery Portal
Web Report Studio
Xythos WebDAV
Stored Process Web Application

The following paragraphs describe some of the lessons learned by the team in the process of installing the BI Platform and migrating the data warehouse to this environment.

IF YOU BUILD IT THEY WILL COME

Plato called necessity the mother of invention. This is certainly true for us. The data warehouse has been "user driven" since its inception. All of the content was created in response to a user's request. In some cases a new report was created to provide the information needed to make administrative decisions. In other situations a manually created report was automated. In still others it was simply a matter of providing a central location that was easily accessible as a repository for reports that already existed. When a user calls every day for two weeks to ask us to run a program for him, it is a sure sign that the program is a candidate to become a stored process that the user can run for himself.

When it came time to retire the HR Vision software, a survey was sent to all HR Vision users to determine which data and reports should be included in the BI environment. This took the guesswork out of this part of the project. One of the disadvantages of the HR Vision software as implemented was the extensive nature of the load process. The load was performed once a month, first on a development system and then in production. There was always a concern that the information was not up-to-date. We were able to automate the load process in BI and load data every night. As a result, the information is never more than 24 hours old. This has been a big hit with former HR Vision users.

One of our concerns was how our users would accept the new system. How do you convince users who are happy with what they have to learn and accept something new? We could just tell them there is no choice but for them to accept it; however, that does not seem very diplomatic. Two factors worked to our advantage in getting users to accept the new system. One was the rapport and respect that had developed with most of the heavy users of the data warehouse. They trusted that what we were doing was for the benefit of everyone. They depended on the data warehouse for essential information used to do their jobs and, in a sense of the word, they didn't want to "bite the hand that fed them." The second factor was the improvements provided by the new system. For most users, the BI platform provided at least one advantage over the old system including more flexibility to customize reports, more recent data, and better organization of content. Now that the BI platform is in place, users continue to create new reports, ask for new data, and request permission to store their documents on the portal.

PROJECT MANAGEMENT CAN BECOME A MANAGEMENT PROJECT

Project management was one of the most troublesome aspects of this project. How difficult could it be to manage 4 people with a clearly defined list of objectives and specific deadlines? In retrospect, "Use the right tool for the job" sounds like very good advice. Management wanted us to track the project with a Gantt chart time line. However, we did not have any project management software for this purpose. Instead, we attempted to create the time line in a spreadsheet using different colored cells to create bars for each person working on a specific task. We never met with management to work out various details of the chart such as how to account for time spent on tasks outside of this project and how to track work on a task beyond the deadline. Not only did our spreadsheet fail to meet management's expectations, it was also cumbersome to work with and time consuming to maintain. There were days before status meetings that we spent more time working on the time line than we did on the project itself. We eventually abandoned the time line and went to a "punch list" type of chart that listed each item that had to be migrated and other tasks that had to be completed. Beside each task was the due date, estimated hours required to complete, and responsible staff. This list was also in a spreadsheet. When an item was completed it was moved from the active sheet to a separate sheet of completed tasks.

Another problem we had with project management was the inability to predict how long certain things would take. This wasn't an assembly line in which we could use a time and motion study to determine how long it would take to build a table. We were trying to predict how long the learning curve would be to figure out something we had never done before and for which we had no training. It was very difficult to feel or appear credible when forced to specify man-hours for these tasks on the time line. It was a guess at best. Some things turned out to be simpler than expected and we beat our estimate. Others proved to be more complex or have unanticipated problems causing us to go way beyond our estimated time. Project deadlines were based primarily on other factors besides the estimated time to complete the tasks. Since we were not renewing our license for HR Vision that deadline was based on the expiration date of the license which we missed. However, we did make it by the end of the grace period. Some deadlines were based on seasonal usage patterns. For example, we wanted to migrate the admissions data by the end of the summer before they started traveling to recruit in the fall. Other deadlines were based on the availability of data. We wanted to be able to load our enrollment data into the new system when it became available on the census date of the spring semester. Since we had other responsibilities in the office, the big question was always, "How much release time for the BI project do we need in order to meet our deadlines?" In the absence of finite man-hour requirements, the real answer was, "As much time as you can give us without compromising the other priorities of the office."

Since we were starting with very little training, no BI experience, and no consulting help, our strategy was to work on

a representative item from each type of content or process. These items served as prototypes. Once we completed and tested all of our prototypes, we could then use what we had learned to migrate similar items from the data warehouse to BI. This was one aspect of project management that actually worked well. After we had developed all of our prototypes, we were at the end of the major learning curve and were better able to provide realistic man-hour estimates. For example, we now know that it usually takes us no more than a couple of hours to convert an application dispatcher program to a stored process.

If we could do this again in an ideal world, we would use project management software to manage the project. We would work out all of the details and parameters before beginning the project so it is clear how the project is to be tracked and reported.

SOME I.T. SKILLS REQUIRED

It would be extremely difficult, if not impossible, to perform a BI installation without Information Technology (IT) resources. There are folders and accounts that must be set up on the servers that house the BI installation. Many of the components of BI are configured with the operating system as services or daemons. These must be stopped and restarted from time to time as part of the BI Platform administration. There are third-party applications such as web servers and scheduling software that must be installed, configured, and maintained. It can be very helpful when troubleshooting to know where to find system logs and logs for the various applications that are part of the BI platform.

There are at least three areas of mastery that need to be represented on the BI administration team: system administration, BI administration, and data knowledge. If all three of these areas are not represented on the team, the missing skill can either be learned or brought in from outside the team depending upon expertise and availability. Ideally, every member of the team would have mastery in all three areas. If only one person possesses all three skills, there is a risk of being crippled should that individual leave the organization. If each skill is provided by a different person, then all must be available and able to collaborate for development or problem resolution. This latter scenario also has certain inefficiencies whose magnitude depends upon the ability of the team members to communicate with and understand one another.

SECURITY: PAY NOW OR PAY LATER

You may have heard the line from the old Fram TV commercial, "You can pay me now or you can pay me later." My conclusion is that this principle applies to the implementation of security on the BI platform as well. You can either invest some time, energy, and perhaps money on security training and planning up front or you can spend a lot of time later trying to solve security problems and changing things that were not set up the way they should have been. Security has been and continues to be one of our greatest challenges in setting up and maintaining the BI environment. The BI metadata repository has its own security infrastructure that must be dealt with in order for the BI platform to function properly. This is in addition to the file and directory security that must be set up on the operating system.

Security on the operating system was relatively straightforward. In our environment, BI files are off limits through the operating system to everyone except staff members in our office. All we had to do was grant permissions to our staff and the appropriate BI accounts, then deny access to everyone else. BI security was another story. Our strategy for BI security was to grant access only to our staff and those who had a business need to access the specific content. In some areas of BI, content inherits permission from the folder. In some areas it does not. There is a complex hierarchy in which permissions are inherited. Inheritance can come from the default setting, system groups, user groups, folders, servers, and Access Control Templates (ACTs). There was no way to see the effective permissions that a user would have on an object and it was often very difficult to trace the source of the permissions. This situation has been improved in BI under SAS® 9.2. I had attended a presentation on BI security by Lisa Frank at SAS® Global Forum. I was intrigued by her use of ACTs to lock down their data. However, I was not convinced that we needed them since their security requirements were somewhat different from ours. We struggled to secure our content. It seemed that every time we tried to set permissions on an object it was either locked down too much or not secure enough. After struggling for several weeks, I ran across a technical paper entitled "Securing SAS® 9 Business Intelligence Content Managed in Metadata." This paper described different strategies for securing BI metadata. We found the strategy that matched our security policies and followed the paper's recommendations to implement it. As it turns out, the ACTs are the key to BI security. We have a separate ACT for each user group. This ACT denies all permissions for the Public group and grants permissions as needed for the specific groups and BI accounts that need access to the content. Our policy has been to grant permissions only at the group level. This allows us to simply add and remove members in the specific group or groups without having to remember every place where a user should have access.

Based on our experience, my recommendation would be to learn as much as possible about metadata security, especially ACTs, before doing any work on a BI installation. Second, plan ahead for user groups and folders to

organize and secure BI content. Folder structure becomes difficult to change once the content is in place and regularly accessed by BI users. Groups and folders should be created based on common usage. In other words, when there is a collection of information used by a specific group of people, this will necessitate a user group in metadata along with one or more corresponding folders in which to organize the content. The security structure in BI mimics the security structure in our campus transactional systems as far as determining specific information to which our users have access.

BACKUP BEFORE YOU HAVE ANYTHING TO BACKUP

Depending upon your definition of a disaster, disaster recovery may not be the only reason for having a backup. We have restored from backup on countless occasions. Once was to make sure a virus had been completely removed. Another was due to a hard drive failure. All of the other occasions were to recover a file that was accidentally deleted, overwritten, or corrupted, or to get a system back to a previous condition where things were working as expected.

When we installed our BI servers and software, we didn't give backups a second thought because our I.T. department maintains a campus-wide backup system that backs up all servers. Then I was looking over the program for the 2007 SAS Global Forum and noticed a presentation entitled "Metadata Backup and Restore with %OMABAKUP". This caused me to question whether our system was truly being backed up. I checked the server logs and sure enough the SAS services have a lock on all the files that make up the metadata repository and they were being skipped by the backup software. We did some research on the %OMABAKUP macro and found some code that we could put into a program that would backup our metadata to a backup folder. It is very easy to backup and restore the metadata repository using this code. We can run the program before making major changes to the metadata. That way, if something goes wrong we can easily undo the changes. This program is also scheduled to run every night just before our system backup occurs. This ensures that our metadata is backed up and provides an archive of several versions through the campus-wide backup system. This process essentially enabled us to turn back the clock at a critical juncture in the project. An incorrect security setting had caused us to start losing user content from the portal. We were able to restore back to a point before the problem began and compare settings to isolate the cause of the problem.

The moral of this story is, "Having a backup doesn't necessarily mean you have a backup." The best practice is to make a backup and test the restore process before loading any data. This way you don't risk wiping out your data with a restore gone bad. Archive this initial copy somewhere safe so that, if necessary, you can go back to your original configuration. If you do your backups right, you can actually toggle back and forth between your current and original metadata configurations in a matter of minutes.

A DEVELOPMENT PLATFORM WOULD BE NICE

We had a development server for HR Vision and the original data warehouse during the last five years of their existence. The intent was that all changes in code would be written and tested on the development server then moved over to production. Unfortunately this is seldom what happened. We were not disciplined enough to take full advantage of the development environment. A test load of the HR Vision warehouses was performed in development each month. All HR Vision patches and upgrades were made on the development server and tested before being put in production. But for the rest of us it was just simpler and easier to make a copy of the code on the production server and make changes to the copy. If we were feeling gutsy we would make the changes in place directly to the production version. With little faith that these habits would change, it was difficult to justify the expense of a development environment for BI. After all, we could do our development on the production server and when we were ready, development would become production.

If we could go back in time with the resources to do whatever we want, we would do it differently. We would have a development server and a production server. We would install and configure the BI software on both servers at the same time. We would do all of our work on the development server. We would not touch the production server again until we were ready to load data and put it in production. There were numerous occasions when a question arose as to whether something came that way or did we make it that way. If we had the two servers as described above we could just look and see how things were initially installed and configured.

We found the first few times we tried to apply hotfixes to the BI software to be a serious challenge. We experienced three or four false starts before we finally figured out how to install the hotfixes without messing something up. If we had a development server we could have learned how to install the hotfixes in development first without affecting production and the learning curve would have been a lot shorter. This would also allow us to apply and test hotfixes in a development environment. As this paper is being written, we lost over half a day of work recovering from and uninstalling two hotfixes that broke our extraction process. We are currently testing a migration from BI on SAS® 9.13 to version 9.2. We are using virtual servers for our development platform where we can perform the migration in development before we attempt it on our production systems.

KEEP A JOURNAL
I have found that often the best way to solve a technical problem is to undo whatever you did immediately before the problem occurred. This is only possible if you know what you did and when you did it. In a project of this magnitude, it is easy to lose track of changes that have been made. Confucius said, "The weakest ink is stronger than the best memory." A journal is one of the best ways to keep track of significant changes and events. The journal should be in a searchable electronic format on a network share that is accessible to all members of the team. The journal should include the date, the person responsible, and a detailed description of what was done and the reason. Journal entries should include but not be limited to configuration changes, hotfix installations, problems resolved, and questions from users. It is also a good practice to log manual data loads including the number of records that were loaded. Don't worry about logging too much information. That would be almost impossible to do. There were a number of occasions when we encountered settings that were different from the default but could not remember why the changes had been made. In too many of those situations we did not have adequate documentation. We learned that it is better to have a good history and not need it than to need it and not have it.

MAKE SURE YOU ARE USING THE LATEST DOCUMENTATION
Besides being environmentally friendly, online documentation is probably the best way to ensure that you are using the latest documentation. Documentation can change fairly rapidly, especially in the early stages of a software's existence. We ran into problems with this on more than one occasion. After installation we were following instructions from a printed copy of the administrator's guide to apply security settings to lock down our BI installation. When we restarted the services the share server would not start up. After some time on the phone with tech support, I was directed to a note addressing this issue located below the chart I had referenced for my security configuration. Unfortunately, this note was not in the version we had printed out even though we had printed it only a couple of months earlier.

In addition to the administrator's guides, we used Course Notes from various BI related training courses to teach ourselves how to use the BI software. We started with a course that is now called "Overview of SAS Business Intelligence and Data Integration Applications". Some of our staff members had attended the course about 15 months prior to the start of our installation. We quickly noticed that many of the screen shots did not match what we were seeing as we tried to work our way through the course. We discovered that the book we were using was at least two versions old already. From that point on, we checked our manuals to see that they were the latest release. Whenever they are available online, we access manuals and administrators guides directly from the SAS website.

DON'T BE A ROBOT
Most interaction with tech support occurs on the phone or through email. In these situations it is very easy to get in the mode of handing off the problem to tech support and waiting for a solution or their next instruction. If they are giving instructions over the phone or are working through a remote connection it is easy to become a robot carrying out their commands without giving any thought to what is being done. We have found it advantageous to remain as engaged as possible in the resolution of a problem. In some cases we were able to solve the problem before we heard back from tech support. We are the eyes and ears for tech support. Their field of vision is often limited to what we enable them to "see". Always provide them with as many details as possible. Don't worry about giving them too much information. If it is more than they need, they will ignore it. It is up to us to help them see the big picture, especially if there is something about our site that is different from a typical installation. We ran into some difficulties on one occasion because we, at the direction of tech support, made some security changes to correct a symptom instead of taking a better look at our overall security configuration. If their instructions don't make sense to you based on your understanding of the situation, don't be afraid to let them know. One of two things will happen. Either they will get more information from the discussion upon which to make a decision or you will learn something about the situation that you did not know. Don't be afraid to ask questions. Some of our best training came from phone sessions with tech support in which they spent some time answering our questions.

YOU STILL HAVE TO UNDERSTAND THE DATA
Regardless of the sophistication, simplicity, or utility of the tools provided, users must still have a good understanding of the data with which they are working. There are several areas in which BI information can be very confusing. This is due to the nature of the data being reported. The better understanding users have of the data, the greater their chances are of producing accurate reports. They will pick up on anomalies when things aren't working as planned. Two examples come to mind. The first user was looking at a cube. He was concerned that the enrollment for an academic year appeared to be over twice what we had reported in the fall. In this particular cube, enrollment for an academic year includes the fall, spring, and summer terms. Once the academic year hierarchy was expanded, the fall numbers were the same as those we reported for the fall term. The second situation occurred with a table listing prospective students. Each prospective student has a model score between 0 and 100 indicating the likelihood that they will enroll at Baylor. It was obvious that there was a problem when a score of 183 showed up on the report. Several fields that would differentiate between students had been removed from the report so that we were now getting duplicate rows. We had not realized that, by default, Web Report Studio aggregates

numeric values and combines duplicate rows. It was adding together the model scores of two students with the same name. By recognizing that a score was out of range, we were able to change the behavior of Web Report Studio to list duplicate rows instead of aggregating them. No amount of technology can substitute for a good understanding of the data.

TRAINING IS AN ONGOING PROCESS

Each time a new content area went into production, we would conduct a training class for the users of this content teaching them how to use the portal and the data. After all areas were migrated we stopped having training. One day we realized that there was no way for new users to get any training. We started conducting two training classes each month in about 9 months a year. The first class each month covers basic use and navigation of the portal and Web Report Studio. The second class of the month covers one of several different content areas such as student data, admissions data, or retention. Classes last about an hour and a half and are hands-on so users can practice on their own. The classes are included in Baylor's professional development program and are advertised to faculty and staff across campus. This has increased awareness of the BI system which has in turn added more users and increased usage of the system. Many users only access the portal once or twice each semester. It is not unusual for them to repeat training classes as a refresher.

CONCLUSION

Our first content area was put into production in just under nine months from the time we started setting up the servers for the BI Platform. Within eighteen months of the start date the old data warehouse was shut down. In that time we had migrated over 25 stored processes, 14 cubes with at least one Web Report Studio report each, 4 tables with information maps and at least one Web Report Studio report each. Two of these information maps were pre-filtered with stored processes. There were also over 20 PDF, html, and Excel documents that were made accessible through the portal, some of which are automatically updated on a daily or weekly basis. In addition, all of the tables and load processes to supply data to these reports were completed. It is our hope that, by sharing these experiences, others who embark on a similar project will have a smoother journey.

REFERENCES

Frank, Lisa, 2009 "At the Controls – An Approach to Security via SAS® Management Console" SAS Global Forum 2009 Proceedings, paper 094-2009. Available at http://support.sas.com/resources/papers/proceedings09/094-2009.pdf

SAS Institute Inc. 2007. *Overview of SAS® Business Intelligence and Data Integration Applications - Course Notes.* Cary, NC: SAS Institute Inc.

SAS Institute Inc. 2006. "Securing SAS® 9 Business Intelligence Content Managed in Metadata." Cary, NC: SAS Institute Inc. Available at http://support.sas.com/techsup/technote/ts750.pdf

CONTACT INFORMATION

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

Faron Kincheloe
Baylor University
One Bear Place #97032
Waco, TX 76798
Phone: (254) 710-8835
Email: Faron_Kincheloe@baylor.edu