Paper 2410-2018

Developing BI Best Practices: Lessons Learned from a Successful BI Implementation

John Barry Taylor, Texas Parks and Wildlife Department; Dan Strickland, SAS® Consulting Services; Alejandro Farias, Texas Parks and Wildlife Department;
Drew Turner, Texas Parks and Wildlife Department

ABSTRACT

When Texas Parks and Wildlife initiated the implementation of a SAS® Business Intelligence (BI) system, it was driven by the business units, and not fully supported by our Information Technology division. Although our implementation team, primarily consisting of staff with backgrounds in biology and accounting, had to learn “on the job” about how to get a complex information technology project off the ground, this project has been successful, and has radically transformed and improved the efficiencies of many aspects of our business processes. There were many lessons learned by our team along the way, and this paper will highlight some of the best practices we developed, as well as several tips and tricks we discovered and would recommend for anyone attempting to implement a SAS® BI installation. Some of the topics covered will include developing implementation and security plans using Best Practices, proven tactics for achieving user buy-in, tips for monitoring and ensuring high levels of server performance, and Best Practices recommendations for providing training and user support.

INTRODUCTION

Our intention in writing this paper is to share some lessons learned during the implementation of a SAS® Business Intelligence (BI) system. Our implementation has been very successful, having grown from under 10 users in the first year to well over 800 users now, and the agency has adopted our SAS® BI system as the host for some key reporting applications, including the entirety of our financial reporting. The impetus for this BI project came from business units of the Texas Parks and Wildlife Department (TPWD), and staff from these business units formed the core of the team building this BI system. While the agency’s IT group provided limited support (such as providing the services of an IT Project Manager, coordinating the set-up of servers in our data center, providing logins for databases we needed to access, and providing some basic data dictionaries for some of the databases as well as some example SQL queries used with the databases), but the majority of the work directly in SAS® was accomplished by people lacking previous IT experience, with some initial help from SAS® consultants. Therefore, the intended audience for this paper includes other SAS® users, especially staff outside of IT, who are just beginning their own BI implementation, or are looking for best practices developed by others to identify potential tweaks to their own existing implementation.

As we began exploring how to use BI within our agency in 2010, we found that while SAS® has what we consider to be excellent documentation on the technical aspects of SAS® BI products, best practices documentation on different models for BI architectures, security and data governance, and methods to enhance the utility and performance of the system for users was lacking. We also found there was very little information that documented actual implementations of BI in other organizations, which we hoped would help us select among alternative models for development of BI best practices. However, we worked with several consultants from SAS® who made helpful suggestions, discussed problems we were facing with platform administrators from other organizations that used SAS® BI, and experimented with different alternative models for elements of our own implementation before settling on the best practices outlined in this paper.

We offer the following lessons learned in creating best practices that were very helpful to the success of our BI implementation. Many of the early lessons learned were less technical in nature, and involved philosophies and approaches that proved helpful in setting up a useful BI architecture, how to market the effort to our prospective user community, stakeholders, and sponsors, and for planning a sustainable support infrastructure. Other lessons learned were more technically-oriented, and we will also address some of these useful lessons that we have applied as best practices to ensure our system performs at a maximum level of efficiency.

We recognize the best practices outlined in this paper are not the ultimate best practices for everyone to follow. Our BI team continues to evolve our best practices by integrating newly discovered methods developed by others; in fact, we believe that striving for constant improvement should be a key best practice everyone should follow to ensure their BI implementation remains effective.

EFFECTIVE PHILOSOPHIES AND APPROACHES TO ACHIEVE CREDIBILITY AND BUY-IN

A PHASED PROJECT SCOPE, BALANCING SCOPE-CREEP AND RISK/REWARD

Having well-defined project goals helps identify project phases and deliverables to minimize misguided fears about project scope-creep. An early understanding that a successful BI implementation should occur in several separate implementation phases, with distinct deliverables in each phase, helped advance the project as a whole.

One of the philosophies we followed to achieve buy-in for our BI project from prospective users was to not to be overly ambitious in what we were attempting to accomplish at any one phase in the process. For example, one of our initial phases was to educate our team who were new to BI on basic BI terminology and concepts, as well as to educate them on ancillary concepts also related to BI, such as data warehousing, effective query logic, etc. and how the different products that make up the SAS® BI product suite could be used to meet differing goals. This was considered a distinct and necessary phase that had to be accomplished before other phases that involved the databases we would surface in the BI system could begin to be addressed. A second example was that we integrated the data from our license sales transactions database as a distinct early phase. This allowed us to use the initial database integration as a test case before integrating other database systems to work out some of the best practices we would follow in integrating other database systems in succeeding phases of the project. Identifying distinct phases which were limited in scope helped manage the expectations of our project stakeholders, and also helped to avoid the overall scope-creep that is so common to these types of projects.

Although we strongly feel that reigning in the ambition for each phase of the project was very helpful in ensuring the overall BI project's success, ironically, one facet that contributed to the success of our BI project stemmed from our willingness to "loosen the reigns" enough to tackle a complex problem -- namely, reporting issues within a newly-implemented financial system -- that was causing large and widespread disruptions for the agency when that strategic opportunity arose.

Our agency had recently implemented a new financial system that was built using a reporting product outside of SAS®. The new financial system had several issues, some of which resulted in inaccurate financial reporting. At the same time this new financial system was implemented, TPWD was progressing our analytical capabilities by investing in our SAS® BI server for the initial purpose of assessing how our customer relationships were associated with the revenues generated by transactions with our customers. Our project was titled the "License Utilization and Revenue Enhancement System" (LURES). Because the main goal of the LURES project was to tie our customer relationships to revenue, the inaccuracy of financial reporting resulted in a big problem for the LURES project. Our BI team concluded that helping resolve the inaccurate financial reporting was not only integral to the success of LURES, but that TPWD would also benefit more generally if we were to also provide an accurate, if alternative, financial reporting system. If successful, it also promised to lend a great deal of credibility to the efforts being expended in the LURES project. We ultimately were able to provide a reporting solution that resolved this agency-wide issue. The strategic benefits of allowing such a seemingly ambitious creep in our scope offered a strategic opportunity that made the risk worthwhile; therefore, we feel a clarified best practice is to seek to balance risk and reward, not just to blindly limit ambition.

A STRATEGICALLY-APPLIED “DISCIPLE APPROACH”

During our collective time as TPWD employees, we have seen many data systems implemented with varying degrees of success, and observed that success or failure of these initiatives has hinged around whether the user community was willing to “buy-in” to its use or not. We believe the success we enjoyed in addressing problems with our new financial reporting system, as well as our BI system in general, stems in large part from the adoption of a philosophy we refer to as the “disciple approach” to introduce SAS® BI to new users. We had seen many examples of new systems and reporting tools being pushed to our prospective user community without having gained their buy-in first, including the reporting tools that had been included with the new financial system, and sensed that the user community was wary of learning new tools.

Therefore, we identified a few select users who were frustrated with their current toolset, and were open to using a new tool like SAS® that had demonstrated proven success in its application. As these users gained trust in SAS® through extensive, ongoing training and careful and attentive customer service, they gained the ability to train and support other new SAS® users, and also became SAS® reporting advocates. This buy-in to using SAS® reporting aroused their colleagues’ curiosity about their improved productivity, prompting their colleagues to inquire about how they could accomplish those same levels of efficiency and productivity. This eventually lead to a “snowball” effect, but we made sure that a sufficient support system was in place before expanding SAS® to each new group of users. Colleagues would eventually self-identify as prospective new SAS® users, and many of them would evolve to serving in a disciple’s role, perpetuating the expanding network of disciples for using SAS®.

In addition to those who self-identified as an interested SAS® user, we also actively sought out users we knew to be influential among their colleagues because of their expertise, seniority, and/or position because we felt they could serve as particularly effective disciples. In a few strategically deliberate instances, we also sought out users who were known to be vocal skeptics of new technologies, business processes, or change in general, and then seized upon opportunities to show them how SAS® could solve problems that were particularly frustrating for them. Getting buy-in with these specific users attracted a great deal of attention and gave a greater air of credibility to what SAS® was enabling our users to achieve.

For example, despite the obvious interrelationship and dependence on financial data, the LURES project had no recognized relationship with the new financial system implementation project. Since the LURES team did not have established authority to assist in the financial system project, we had to approach staff involved with the financial project informally and carefully to avoid the perception of being intrusive. We approached a staff member recognized as having deep technical knowledge of the financial business function, and who had been frustrated with using the query tools offered by the new financial system. This staff member had considerable expertise in the use of Excel, so we offered to install the SAS® Add-In for Microsoft Office on his computer. He began data explorations that identified some errors in some of the tables within the financial system. He subsequently began using SAS® Enterprise Guide to build queries from scratch, reverse-engineer existing report queries that appeared to contain errors, and identify needed fixes to the underlying SQL code. Additionally, he explored and implemented several new architectural constructs that greatly increased the speed of query executions as compared to the pre-existing non-SAS® query tool and architecture. He began to share his SAS® financial reports with colleagues, as well as his supervisors, and this offered a proof of concept that SAS® could be used to remedy many of the financial reporting problems that had frustrated the user community. As his colleagues saw the benefit of using SAS®, the LURES project also progressed, and the BI user base began to grow.

VALUE-ADDED DATA PRODUCTS

Two of the leaders of our BI project had previous training and/or experience in marketing within the field of agricultural sciences, and one of the influential concepts in that field is the focus on providing consumers “value-added” products. In much the same way that an agricultural product may need to undergo additional processing steps beyond harvest to enhance its quality, usability, and accessibility before consumers may be attracted to it, we contend that similar processing steps can enhance the attractiveness of data products for consumers.

Although many think of marketing as primarily addressing the promotional aspects of a product, one should instead recognize marketing should address what is known as the “marketing mix”, which incorporates the five ”P”s of marketing: Product, Price, Place, People, and finally, Promotion. Value-added concepts recognize the need to address the total marketing mix.

To give an example in the agricultural field, over the past few decades, research has proven it is technologically feasible to produce freshwater shrimp in aquaculture enterprises; however, the financial feasibility of these enterprises was lacking until producers began to concentrate on several additional steps that added value to shrimp-based products in the consumer’s minds, including:

- Improving the quality and conformity of the product (e.g., establishing proper handling and processing techniques)
- Improving packaging and ease of use by the consumer (e.g., producing individually quick-frozen shrimp in re-sealable plastic bags that can be quickly prepared at home)
- Expanding the accessibility of the product in the marketplaces consumers frequent (e.g., opening distribution channels that can supply packaged shrimp to popular grocery chains)
- providing educational opportunities on the advantages offered by the product, and how to use it, to raise awareness among consumers who are unfamiliar with the product (e.g., in-store tasting and cooking events, clearly-labeled instructions and menus on packing, social media and other online media)

This focus on understanding what the customer really wants from a product, how to best enhance its usability and accessibility, and focus on educationally-based awareness of the product is a philosophy we applied to our all of data products.

When we first envisioned the implementation of SAS® BI for our LURES project, we assumed the majority of our users and the majority of the content in LURES would be consumed with the browser-based BI tools such as the SAS® Information Delivery Portal, SAS® BI Dashboard, and SAS® Web Report Studio. However, we discovered that much of the success we had in achieving buy-in among our user community was because we offered them the option to use the SAS® Add-In for Microsoft Office as an interface with SAS® and the BI content. Most of our users were familiar and comfortable with the Excel environment; there seemed to be a psychological advantage for users of Excel, apparently because they felt they were only learning a new aspect of Excel and not an entirely new client.

We’ve read many BI articles by experts who complained that Excel is a poor analytical tool, primarily due to the tendency for communities of Excel users to have multiple versions of spreadsheets that are intended to do the same thing, but that have different formulas, underlying data, etc. This results in “many versions of the truth” spread throughout the organization. We found that the combination of a SAS® BI server, and centralized, well-designed, and properly-curated data products, such as SAS® Information Maps, Stored Processes, and SAS® data tables stored in consumer-friendly data marts containing thoroughly vetted, cleaned, and properly-formatted data, helped achieve “one version of the truth”.

By offering our users this curated BI reporting data repository, and extensive and ongoing training on the proper use of these data products, we have achieved one version of truth throughout the organization, while offering our users the benefit of using the SAS® Add-In for Microsoft Office client with which most of them are most comfortable and productive.

ON-GOING EDUCATION

Our team had limited financial resources for training, consequently pushing us to find cost effective ways to learn. Although we did take formal courses from the SAS® Institute that were very useful, self-training is another best practice we employed to help in developing the knowledge base for each content delivery tool at a fraction of the cost of attending formal training. Finding self-teaching media such as SAS® course notes purchased form the SAS® Institute, YouTube how-to videos, User Group papers on LexJansen.com, as well as other repositories such as SASCommunity.org and other user's blog sites allowed us not only to teach ourselves, but develop the skills to properly train disciples.

To get started, we had to become conversant in the terminology used to describe the elements of BI, and we started simply with the book, "Business Intelligence for Dummies" (Scheps, 2007). In addition to describing what was meant by BI-specific terms like OLAP and other technical terms that were new to us, it included suggestions for some helpful general philosophies and approaches for achieving buy-in by ensuring a BI implementation is relevant and useful to the user community. If you are new to the world of BI, we'd recommend this book as a primer.

As our agency SAS® user base continued to grow, we developed an in-house user group and provide regular trainings on SAS® design and analytical techniques. Providing learning opportunities to even our more-experienced users facilitates the effectiveness of our "disciple approach" to continue to expand our user base and their reliance on the SAS® BI system.

DEVELOPING BEST PRACTICES FOR SECURITY

DETERMINE A PRACTICAL RESPONSIBILITY STRUCTURE FOR DATA GOVERNANCE

When this project began in 2010, those of us on the in-house BI SAS® implementation team at TPWD had never heard of the term "Data Governance", but we realized that our largest risk for failure was implementing a BI system that gave insecure access to the data owned by the organization. A failure such as that could quite literally have ended the project. Since the main focus of LURES is to assess all the customer-based transactions residing in a multitude of databases and relate them to the underlying financial system that tracked revenue generated during those transactions, there is highly sensitive personally identifiable information stored in those databases. As a result, proper security to protect that data was of utmost importance.

We also recognized that the administrators responsible for any business function should be considered "Data Owners", and that Data Owners could rely on other, more technically-oriented staff whom we designated "Data Stewards", for the day-to-day responsibility and technical knowledge needed to maintain the data. It was important to recognize that these data owners and stewards, who are in business units, rather than IT, have the ultimate authority over any decisions based around security for their particular data, while also adhering to security protocols established by our IT group. So, our first step was to survey the transactional databases we wanted to include within the LURES project and determine who should be the Data Owners and Data Stewards for each. We then met with these owners and stewards individually to explain the purpose and goals of our BI project and get feedback on how they would like us to use the data, as well as address any reporting needs for their specific business function outside of the core of the LURES project. If they had an interest, we invited them to join the LURES project as an active team member.

Involvement of the data owners and stewards was helpful in an additional way. Documentation such as data dictionaries, entity-relationship diagrams, etc. was lacking for many of the databases we intended to incorporate into our BI project. However, based on information and example SQL queries provided by the data owners and stewards, we were able to construct missing documentation. Reverse-engineering of these SQL queries also enabled us to construct accurate entity-relationship diagrams and inform the fleshing out of data dictionaries in several cases where previous documentation was nonexistent.

PLAN FOR FLEXIBILITY

We understand that one of the central goals of most security plans is standardization of security protocols, and agree that there are several important reasons this is a desirable goal, including:

- Consistency of development across the enterprise
- Improved comprehension
- Understanding of security goals and protocols
- Implementation and maintenance

Simplicity among security protocols also offer similar advantages. However, some of the data systems surfaced require more complex security structures than others. In the interest of practicality, security plans must allow for some flexibility where required. A security model should not limit a user's approved access to data, and should be simple, standardized, but most importantly, flexible in securing sensitive data.

DON'T REINVENT THE WHEEL

When we began developing our BI system, the agency's IT group was also in the early stages of developing a comprehensive security plan to meet new state-mandated security standards applicable to all state agencies. Although the new protocols our agency would seek to implement had not yet been fully established, there were some previously-existing protocols, and we knew enough about the oversight agency's proposed standards to make some "educated guesses" as to what any new protocols would eventually entail. We built our security framework keeping these established, as well as prospective, security standards in mind. That minimized the risk of being forced to make major changes to our security framework in response to any security audits, while avoiding delays caused by waiting for new protocols to established. Our BI implementation has been in place for over six years, and we have not had to make any changes up to this point. Because we were willing to be proactive in establishing our best guess at a workable security model, instead of waiting for new protocols to be approved, we were able to go into a production environment much faster. If conflicting protocols arise in the future, we believe we will be able to make minimal changes to accommodate them. More importantly, because our security model is currently effective and being used throughout the agency, it is likely to influence any new protocols that may be considered.

SAS® METADATA OFFERS A SECOND LAYER OF SECURITY ON TOP OF EXISTING SECURITY

It is important to recognize that the security offered through the SAS® Metadata server sits on top of the security that is already in place on existing data systems that are surfaced through the BI environment, and the SAS® layer of security augments, rather than replaces, operating system security and database security. SAS® adds an additional, improved layer of security that goes above and beyond our organization's security model. SAS® has extensive documentation on the security systems that may be implemented within their software, but the information provided in this document references how we are specifically using the security features in SAS® to meet the goals of TPWD's security model.

SECURITY CREDENTIALS EMPLOYED ON THE SAS® PLATFORM

The SAS® server platform consists of several SAS® software servers that are all working together. It is composed of the SAS® Metadata Server, SAS® OLAP Server, SAS® Workspace Server, SAS® Pooled Workspace Server, SAS® Stored Process Server, SAS® Object Spawner, SAS® Data Step Batch Server, SAS® Java Batch Server, SAS® Logical Table Server, and SAS® Content Server. These servers require proper credentials to allow a user to be authenticated as an accepted user of the SAS® environment. It's possible for credentials to be supplied either from a user's submitted program or from the SAS® Metadata, but we encourage users to work through the Metadata, and not embed credentials in programs. In addition to the credentials and authentication procedures required by the SAS® environment, there are various databases that are being incorporated within the SAS® BI system which require separate credentials and authentication procedures.

UNDERSTANDING USER METADATA

The main server controlling access permissions is the SAS® metadata server, and at TPWD, we have integrated Active Directory with the SAS® Metadata. During the initial login into the SAS® system, the metadata server uses the TPWD Domain's Active Directory registration to verify that a user's username/password allows them to log into the metadata server as an authorized TPWD user. If the user does not have Active Directory credentials they will not be able to access the TPWD SAS® system, because it resides behind the TPWD firewall. Additionally, the user must have credentials registered within the SAS® Metadata server to authenticate as a user with recognized access privileges to any specific data or reports, and we grant the user access to specific data by membership in groups.

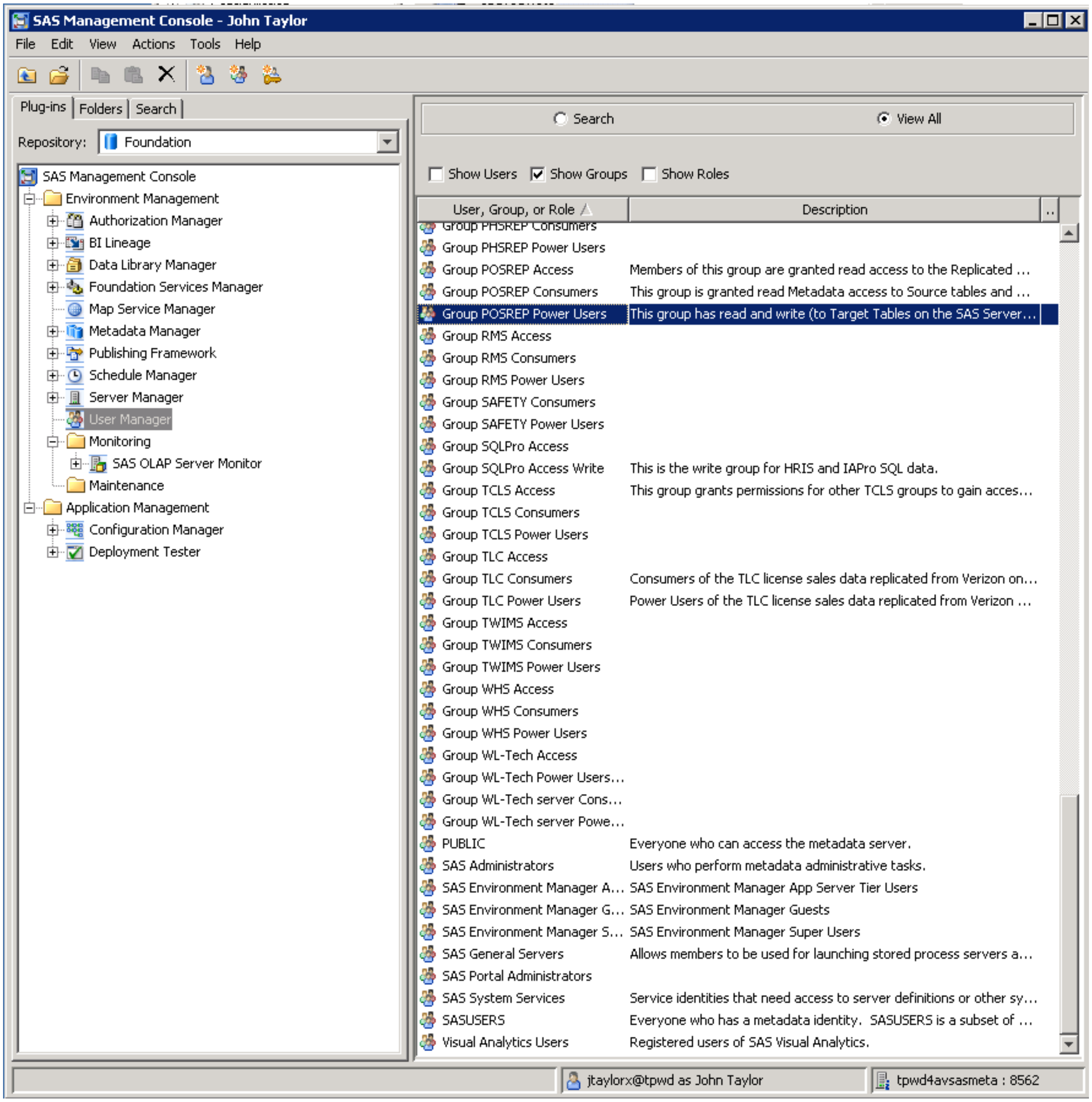
At the most basic level of access to the data surfaced by the TPWD SAS® BI system, users must have Active Directory credentials on the TPWD domain; if they do not also have an individual identity on the SAS® Metadata server, they will authenticate within the SAS® system at a "Public" level of access rights. However, we currently have Public users' permissions set to deny access to all of our data, and to the SAS® server's analytical engine.

One step above the Public level is the "SASUser" level. This level requires them to have an individual profile registered in the SAS® Metadata and they are also authenticated against TPWD's active directory. This is the minimal level at which a user can currently access our SAS® server's analytical engine; a SASUser is able to analyze data from their local drives in the SAS® Add-In for Microsoft Office or the SAS® Enterprise Guide client, for example A SASUser is also able to be placed in groups that are granted specific permissions to access databases integrated with the SAS® Metadata server by accessing metadata objects (servers, folders, libraries, Information Maps., Stored Processes, etc.). If a database requires individual logins, the SAS® Platform Administrators can add a user's database credentials to the individual's SAS® Metadata profile which will allow them access to that specific database. If a database is set up to allow users to share access credentials using a shared login account, SAS® Platform Administrators set up those credentials within the group-level Metadata profile.

UNDERSTANDING GROUP METADATA

The next level of access privileges is controlled by membership to groups. We currently have a minimum of two groups for each database; more groups, with differing access levels, can be created for a specific database as needed. Users only have access to data for which they are members of specific databases' "Consumers" or "Power Users" groups. We have granted "Consumers" groups the right to read metadata for database source tables for which they have been granted access, which allows them to run Stored Processes and Information Maps, but they are not granted the right to read the raw data tables. For a specific database, "Power Users" groups have all the access rights of Consumers, but Power Users are also granted additional privileges that allow them the ability read raw data tables, to create BI content, and save content to the Metadata folders. Therefore, BI content and report developers are given "Power Users" memberships.

As with individual users, groups also have Metadata profiles. As stated above, TPWD has a minimum of two groups per database server. These groups are typically named with the convention “Group [database name] Consumers” and “Group [database name] Power Users” (see Figure 1). Roles, which control how each client product (e.g., Enterprise Guide, Add-In for Microsoft Office, etc.) can be used, are also assigned to each group.



□ Show Users ☑ Show Groups □ Show Roles

User, Group, or Role	Description
Group PHSREP Consumers	
Group PHSREP Power Users	
Group POSREP Access	Members of this group are granted read access to the Replicated …
Group POSREP Consumers	This group is granted read Metadata access to Source tables and …
Group POSREP Power Users	This group has read and write (to Target Tables on the SAS Server…
Group RMS Access	
Group RMS Consumers	
Group RMS Power Users	
Group SAFETY Consumers	
Group SAFETY Power Users	
Group SQLPro Access	
Group SQLPro Access Write	This is the write group for HRIS and IAPro SQL data.
Group TCLS Access	This group grants permissions for other TCLS groups to gain acces…
Group TCLS Consumers	
Group TCLS Power Users	
Group TLC Access	
Group TLC Consumers	Consumers of the TLC license sales data replicated from Verizon on…
Group TLC Power Users	Power Users of the TLC license sales data replicated from Verizon …
Group TWIMS Access	
Group TWIMS Consumers	
Group TWIMS Power Users	
Group WHS Access	
Group WHS Consumers	
Group WHS Power Users	
Group WL-Tech Access	
Group WL-Tech Power Users…	
Group WL-Tech server Cons…	
Group WL-Tech server Powe…	
PUBLIC	Everyone who can access the metadata server.
SAS Administrators	Users who perform metadata administrative tasks.
SAS Environment Manager A…	SAS Environment Manager App Server Tier Users
SAS Environment Manager G…	SAS Environment Manager Guests
SAS Environment Manager S…	SAS Environment Manager Super Users
SAS General Servers	Allows members to be used for launching stored process servers a…
SAS Portal Administrators	
SAS System Services	Service identities that need access to server definitions or other sy…
SASUSERS	Everyone who has a metadata identity. SASUSERS is a subset of …
Visual Analytics Users	Registered users of SAS Visual Analytics.

jtaylorx@tpwd as John Taylor

tpwd4avsasmeta : 8562

Figure 1. The User Manager utility in SAS® Management Console, displaying an example of Groups that have been set up for the TPWD SAS® Server.

At TPWD, we assign roles to the group metadata profiles and not to individual users’ profile. This enables us to manage the roles more effectively. Permissions are also assigned to each group according to the permission levels needed for each specific Metadata folder. We can also place Access Control Templates (ACT) on groups which allow us to efficiently set permissions and access through the use of these templates.

Following is a list of permissions used in the access control templates:

- **ReadMetadata**: Ability to see a metadata object
- **WriteMetadata**: Ability to add, modify, and delete metadata
- **WriteMemberMetadata**: Ability to add, modify, and delete metadata objects in folders
- **CheckInMetadata**: Ability to check metadata back to foundation or other repository from a project repository
- **Read**: Ability to read data
- **Write**: Ability to modify existing data
- **Create**: Ability to add new data.
- **Delete**: Ability to delete data
- **Administer**: Ability to administer SAS® OLAP Server and SAS® Table Server

ORGANIZATION OF DATA FOLDERS IN THE SAS® METADATA

TPWD’s data folder and group level architectures are designed around the database systems which we are incorporating within the BI system. When we began designing our architectures, we considered structuring these architectures around our organizational structure (e.g, TPWD Division memberships such as Inland Fisheries Division, Law Enforcement division, etc.) or around job function roles as we’d seen in some SAS® BI course examples (e.g., Finance, Payroll, etc.). After considering issues we’d face in dealing with the potential for having to modify architectures based on structures like these, which are somewhat arbitrary and subject to periodic reorganization, we decided to structure our architectures around the structure of our database systems. We have found this to be a very effective best practice for several reasons, including:

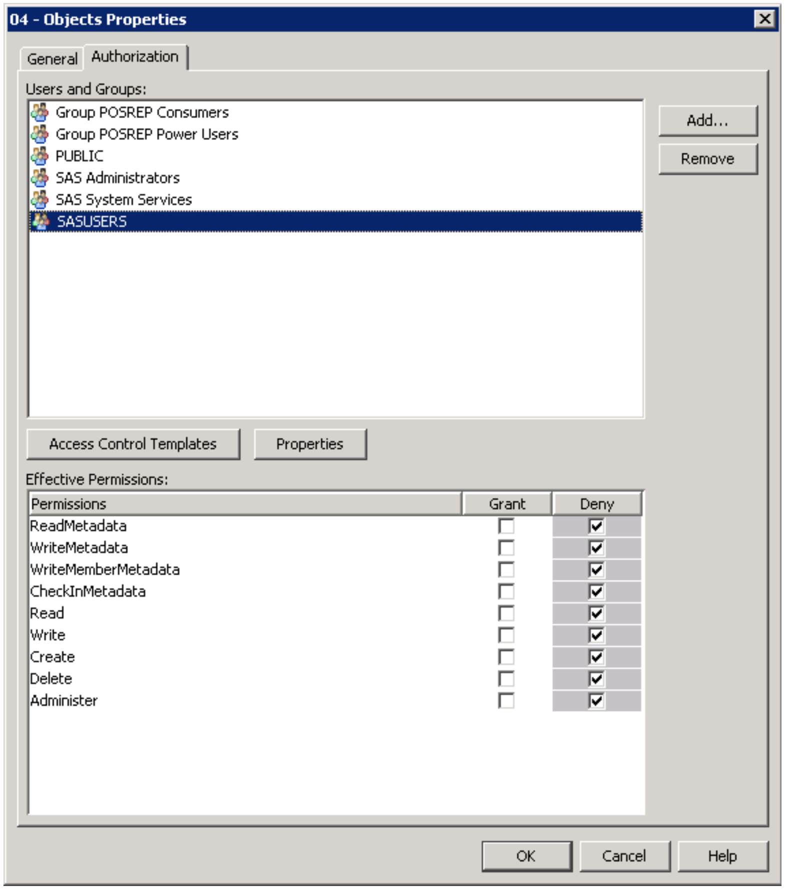
- Provides a logical structure that reflects how users typically interact with each system
- Simplifies administration of Metadata and SAS® BI Platform Administration tasks.
- Minimizes impacts when database systems are migrated to upgraded servers

For each database server, we have set up a minimum of 5 subfolders, with permissions set on each folder:

- Folder “01 - Source Tables” contains the libraries of registered tables of metadata for that database schema
- Folder ”02 - Jobs” contains scheduled SAS® Data Integration Studio batch programs
- Folder “03 - Target Tables” contains the SAS® data tables formed from the jobs, and also surfaces data stored in the SAS® data warehouse
- Folder “04 - Objects” contains the approved and published Information Maps and Stored Processes that have been approved by the databases’ data owners and stewards for Consumers group use, after undergoing proper review and verification
- Folder “05 -Stage” contains objects that are either still under development or are awaiting approval

Following is a typical example of permissions set on each of these folders for a specific database system.

Permissions are set to ‘Deny’ for Public and SASUsers permissions are set to ‘Deny’ on all permissions (see Figure 2).



Users and Groups:

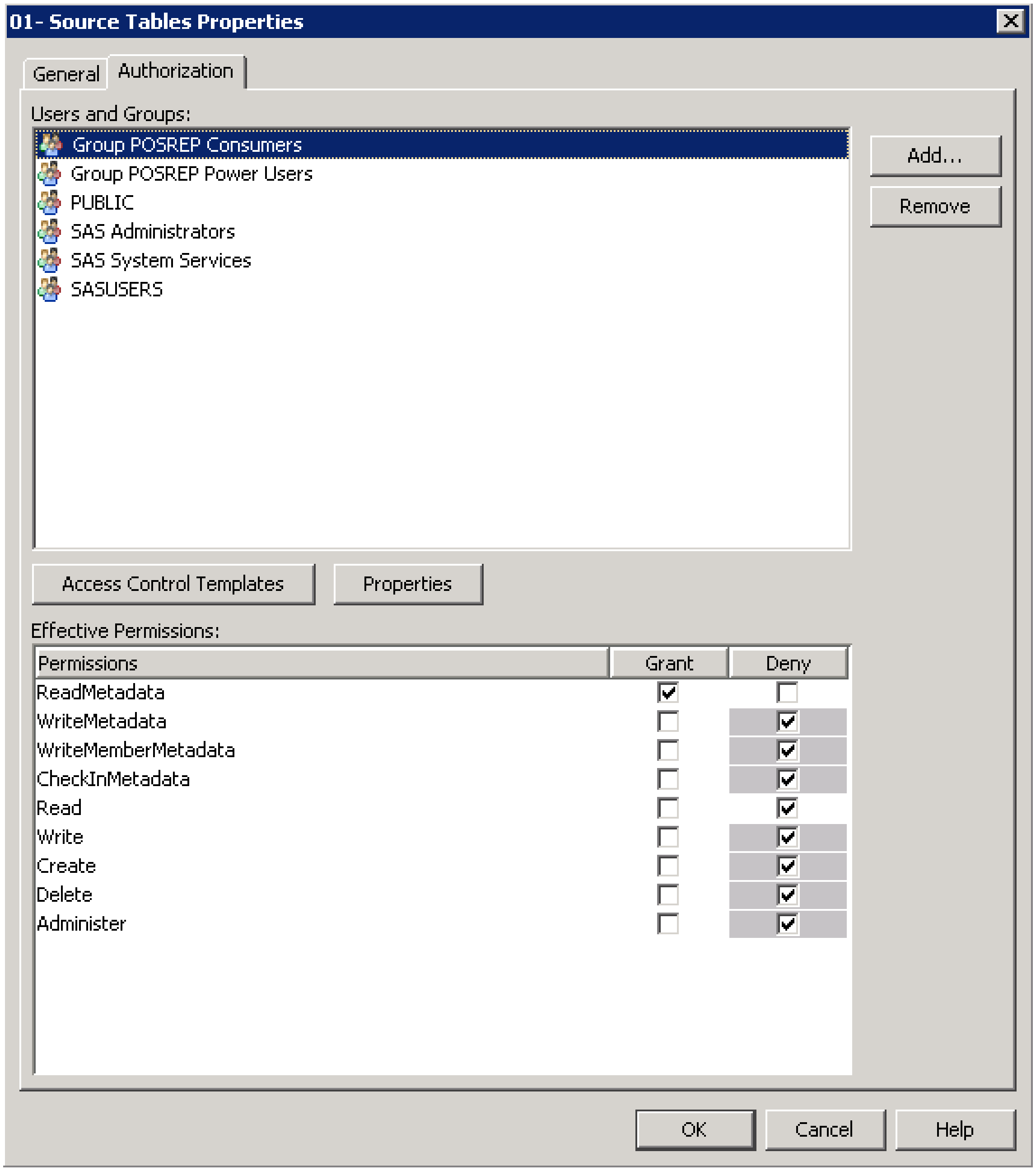
- Group POSREP Consumers
- Group POSREP Power Users
- PUBLIC
- SAS Administrators
- SAS System Services
- SASUSERS

Effective Permissions:

Permissions	Grant	Deny
ReadMetadata	□	☑
WriteMetadata	□	☑
WriteMemberMetadata	□	☑
CheckInMetadata	□	☑
Read	□	☑
Write	□	☑
Create	□	☑
Delete	□	☑
Administer	□	☑

Figure 2. An example of the denied permissions for SASUsers in the typical folder security model at TPWD.

The “Consumers” group is only given Read Metadata permissions for the Source Tables folder (see Figure 3), but Consumers have Read Metadata as well as Read permissions for the target tables and Objects folders. This set of permissions allows our Consumers access to the curated and properly-vetted SAS® tables, Information Maps, and Stored Processes, but disallows them access to the raw Source Tables.



Users and Groups:

- Group POSREP Consumers
- Group POSREP Power Users
- PUBLIC
- SAS Administrators
- SAS System Services
- SASUSERS

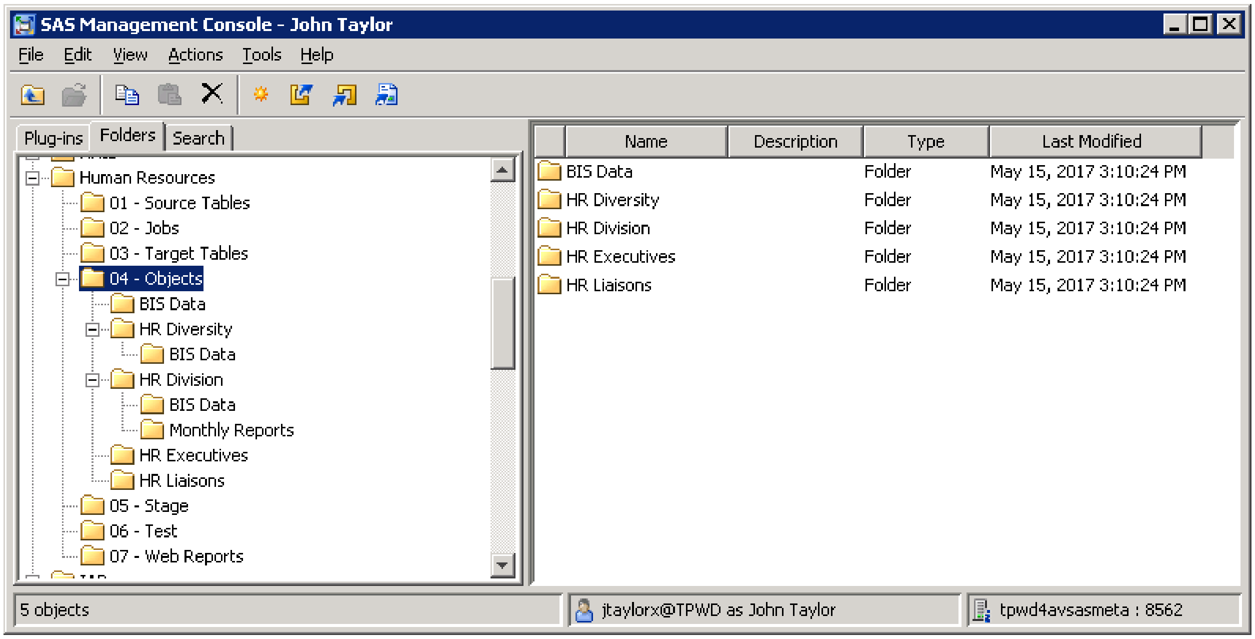
Effective Permissions:

Permissions	Grant	Deny
ReadMetadata	☑	□
WriteMetadata	□	☑
WriteMemberMetadata	□	☑
CheckInMetadata	□	☑
Read	□	☑
Write	□	☑
Create	□	☑
Delete	□	☑
Administer	□	☑

Figure 3. An example of the permission levels for "Consumers" on the Source Tables folders for databases within the TPWD SAS® BI System.

We grant Power Users all of the Metadata-related permissions as well as Read permissions for all of the subfolders of a specific database system, which allows them to access raw Source Tables, and create content for Consumers and Testers, and to schedule ETL jobs that create the tables in the SAS® data warehouse, which is surfaced under the Target Tables subfolder. We also ensure that a particular Power User may have Power User status for only one or more database system that the Data Owner and Data Stewards for that system have approved, and that user may otherwise have no access, or only consumer-level access, to other database systems.

As mentioned above, more complexity is required for some database systems, and we utilize more complex group and folder structures in those cases. For example, the Data Steward for our human resources (HR) data identified four different levels of consumers for whom data and reports would be surfaced. The Data Steward did not want members of any one consumer group to see the content for another group; therefore, we set up four separate HR “Consumer” groups, and created four separate corresponding Metadata folders (HR Diversity, HR Divisions, HR Executive, and HR Liaisons) under the Objects folder (see Figure 4). Each of these four groups are allowed to see different data that require different levels of security.


SAS Management Console - John Taylor

Name	Description	Type	Last Modified
BIS Data		Folder	May 15, 2017 3:10:24 PM
HR Diversity		Folder	May 15, 2017 3:10:24 PM
HR Division		Folder	May 15, 2017 3:10:24 PM
HR Executives		Folder	May 15, 2017 3:10:24 PM
HR Liaisons		Folder	May 15, 2017 3:10:24 PM

5 objects | jtaylorx@TPWD as John Taylor | tpwd4avsasmeta : 8562

Figure 4. Screen shot of the folder structure for TPWD’s Human Resources Metadata.

SHARING LOGINS AMONG GROUPS

When we first began setting up metadata groups to provide different levels of access, we assigned login permissions to consumer groups, then added the Power Users group as a member of the Consumer group in order to share login credentials. Making groups members of other groups negates the need to have a separate login credential for each group. Understandably, our database administrators want to limit the number of logins they have for each database. Later, at the suggestion of one of our SAS® consultants, we began creating groups with the naming convention “Group [database name] Access”, to which the a database login was assigned; the members of that group are the Consumers, Power Users, or other more specialized groups (if any) associated with that database, allowing all of those member groups to share a single login.

PROTECTING CONSUMERS FROM TROUBLE

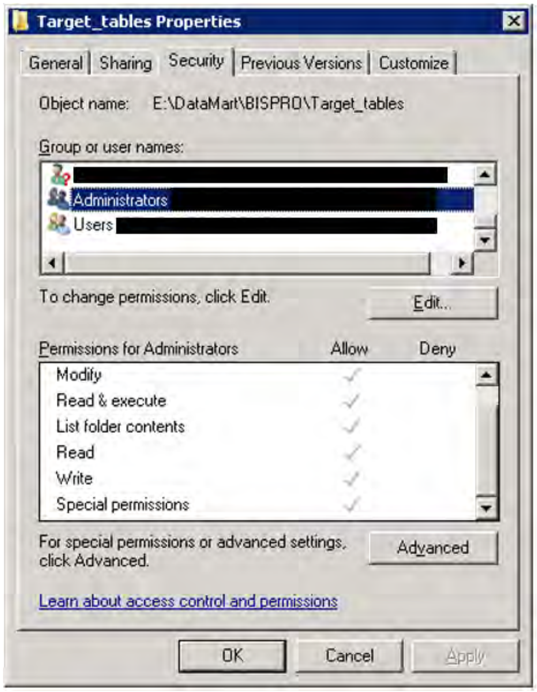
Like most databases, many of ours were not designed to be used directly by end users. Usually, it is envisioned that the end-users will access the data indirectly by running canned reports that had been prepared for specific, previously-defined reporting needs. Although analysts outside the IT group are often asked to perform ad hoc analyses on the databases because the canned reports do not meet the needs of some previously-undefined report, the raw databases typically are constructed in such a way that it is difficult for an analyst outside of the database administrators and/or programmers who built the data system to perform. One of the missions of the BI project was to provide enhanced ad hoc analytical capabilities to analysts in the business units of TPWD. Some of the problems we faced in providing ad hoc analytical capabilities included:

- Many of the data tables used field name nomenclature that is somewhat cryptic, and data dictionaries were not readily-available to analysts.
- Most of the databases were complex relational databases comprised of many tables and complex SQL joins are required to conduct most queries.
- Many of the fields were stored in formats that were not intuitively obvious to analysts (e.g., dates being stored as a date-time field, currency values being stored pennies instead of dollars, currency values all being positive and required reference to a separate field to determine if it should be a debit or credit, etc.).

Therefore, we realized we needed to protect consumers of our data from making errors in their analyses because data were not set up as they might have assumed, and we wanted to provide them with cleaned and curated data products that were designed to be consumed by analysts lacking advanced knowledge of database construction and/or extensive SQL programming knowledge. To do this, we wanted to provide access to the data products in folders separate from the raw source tables, and to deny access to the raw source tables by consumers. Within the consumer-oriented folders, we provided access to cleaned and properly-vetted data tables stored in the SAS® data warehouse which had been created from the raw source data, as well as Information Maps, OLAP Cubes, and Stored Processes. Granting Consumers only “Read Metadata” privileges, while denying all other privileges to the folders containing raw source database tables allowed Consumers to run Information Maps, OLAP Cubes, and Stored Processes, but prevented them from being able to read raw source tables.

MANAGING THE SAS® DATA WAREHOUSE

Our SAS® BI platform runs on a Windows operating system, and our data warehouse is also stored in Windows OS folders on the same server as our Compute tier (two other servers contain the Mid-Tier and the Metadata tier). We created a Windows OS folder named “DataMart”, where all the SAS® tables in our Data Warehouse are stored, with separate folders for each database listed under the Data Mart Folder. Another folder is named “CodeMart”, and the SAS® code for Stored Processes is written there, in separate sub-folders for each database. Users who are not granted Windows OS-level “Write” and “Modify” permissions either as an individual or as a named member of an Active Directory Group cannot write to the Windows OS folders. Only Members of the “Administrators” Active Directory group or named users who we are designated as an ETL developer by the Data Owners and Stewards have write permissions that allow them write SAS® data to the tables in the Data Mart (see Figure 5).



Object name: E:\DataMart\BISPRO\Target_tables

Group or user names:

- [redacted]
- Administrators [redacted]
- Users [redacted]

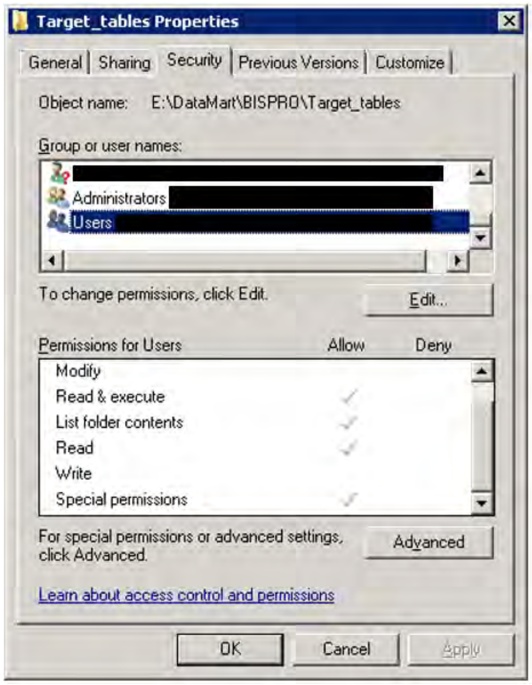
To change permissions, click Edit.

Permissions for Administrators	Allow	Deny
Modify	✓	
Read & execute	✓	
List folder contents	✓	
Read	✓	
Write	✓	
Special permissions	✓	

For special permissions or advanced settings, click Advanced.

Figure 5. An example of the Windows OS permissions granted to Administrators in the SAS® Data Mart.

Users are granted “Read and Execute”, “List Folder Contents”, and “Read” permissions at the Windows OS-level, which allows all of our users to read data from the Data Mart, if they have the proper SAS® Metadata permissions (see Figure 6).



Object name: E:\DataMart\BISPRO\Target_tables

Group or user names:

- [redacted]
- Administrators [redacted]
- Users [redacted]

To change permissions, click Edit.

Permissions for Users	Allow	Deny
Modify		
Read & execute	✓	
List folder contents	✓	
Read	✓	
Write		
Special permissions	✓	

For special permissions or advanced settings, click Advanced.

Figure 6. An example of the Windows OS permissions granted to Users in the SAS® Data Mart.

RECOMMENDATIONS FOR IMPROVING PERFORMANCE AND EFFICIENCY

CHOOSING THE MOST-APPROPRIATE TOOL TO SURFACE DATA

Choosing among SAS® Stored Processes, Information Maps, data tables, OLAP Cubes, Web Reports, or Dashboards (or any combination of two or more) as the most appropriate venue to surface data can be difficult. There is no easy answer for choosing which method to use to build reports. There are typically advantages and disadvantages with each data product that may be lessened or magnified, depending on the situation.

For one example, we find Information Maps are an efficient method for surfacing data when the underlying processing runs quickly, and using them can save on storage space in the Data Mart. However, since the processing steps for an Information Map are only executed on demand (akin to SQL views), for large datasets, or for when the underlying table joins and other processing steps required to execute a particular query require substantial processing time, we find that as an alternative, scheduling ETL processes jobs to build data tables (akin to SQL materialized views) are more efficient for surfacing data.

Having a thorough understanding of each content delivery tool and a thorough understanding of the users' reporting requirements allows the BI content developer to determine the best tool (or mix of tools) to produce the specific report. This knowledge takes time and experience to acquire. A thorough description of all the considerations of selecting the most appropriate methodology is beyond the score of this paper. We plan to submit a follow-up best practices paper to detail the many lessons we learned on this topic.

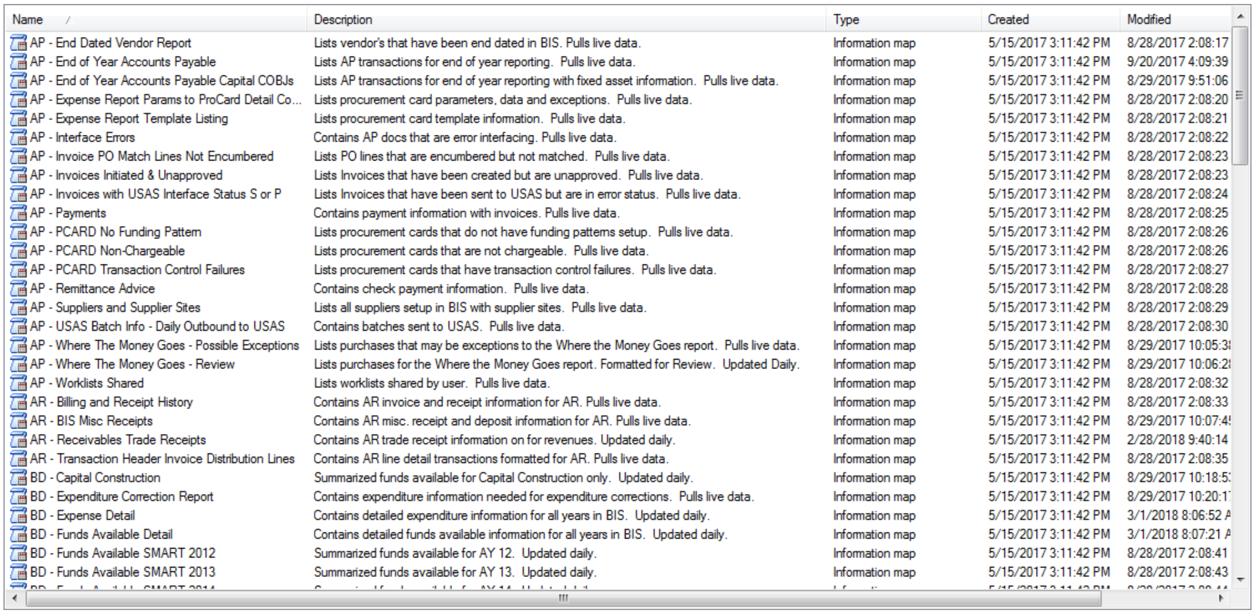
ESTABLISH A NAMING CONVENTION FOR CONTENT

Another best practice our BI team developed was to simplify report selection for our users by providing a report catalog. Reports are first classified by group (see Table 1 for an example), then given a specific name with a short description of what the report achieves and whether the underlying data is live or updated daily. Cataloged reports allow users to scan all reports by function, then by name, then by output.

Group	Description
AP	Accounts Payable
AR	Accounts Receivable
BD	Budget
EP	Employee
FA	Fixed Asset
GA	Grants
GL	General Ledger
GN	General
LT	List
PO	Purchase Order
RN	Reconciliation
RV	Revenue
US	USAS

Table 1. Example functional group naming convention for assigning acronyms used in report names.

Users can quickly identify what they are looking for and whether the report outputs suit their specific needs. Each aspect of the catalog (group, name, and output description) should be maintainable and easy-to-comprehend for your users. Figure 7 demonstrates an example of our report naming convention.



Name	Description	Type	Created	Modified
AP - End Dated Vendor Report	Lists vendor's that have been end dated in BIS. Pulls live data.	Information map	5/15/2017 3:11:42 PM	8/28/2017 2:08:17
AP - End of Year Accounts Payable	Lists AP transactions for end of year reporting. Pulls live data.	Information map	5/15/2017 3:11:42 PM	9/20/2017 4:09:39
AP - End of Year Accounts Payable Capital COBJs	Lists AP transactions for end of year reporting with fixed asset information. Pulls live data.	Information map	5/15/2017 3:11:42 PM	8/29/2017 9:51:06
AP - Expense Report Params to ProCard Detail Co...	Lists procurement card parameters, data and exceptions. Pulls live data.	Information map	5/15/2017 3:11:42 PM	8/28/2017 2:08:20
AP - Expense Report Template Listing	Lists procurement card template information. Pulls live data.	Information map	5/15/2017 3:11:42 PM	8/28/2017 2:08:21
AP - Interface Errors	Contains AP docs that are error interfacing. Pulls live data.	Information map	5/15/2017 3:11:42 PM	8/28/2017 2:08:22
AP - Invoice PO Match Lines Not Encumbered	Lists PO lines that are encumbered but not matched. Pulls live data.	Information map	5/15/2017 3:11:42 PM	8/28/2017 2:08:23
AP - Invoices Initiated & Unapproved	Lists Invoices that have been created but are unapproved. Pulls live data.	Information map	5/15/2017 3:11:42 PM	8/28/2017 2:08:23
AP - Invoices with USAS Interface Status S or P	Lists Invoices that have been sent to USAS but are in error status. Pulls live data.	Information map	5/15/2017 3:11:42 PM	8/28/2017 2:08:24
AP - Payments	Contains payment information with invoices. Pulls live data.	Information map	5/15/2017 3:11:42 PM	8/28/2017 2:08:25
AP - PCARD No Funding Pattern	Lists procurement cards that do not have funding patterns setup. Pulls live data.	Information map	5/15/2017 3:11:42 PM	8/28/2017 2:08:26
AP - PCARD Non-Chargeable	Lists procurement cards that are not chargeable. Pulls live data.	Information map	5/15/2017 3:11:42 PM	8/28/2017 2:08:26
AP - PCARD Transaction Control Failures	Lists procurement cards that have transaction control failures. Pulls live data.	Information map	5/15/2017 3:11:42 PM	8/28/2017 2:08:27
AP - Remittance Advice	Contains check payment information. Pulls live data.	Information map	5/15/2017 3:11:42 PM	8/28/2017 2:08:28
AP - Suppliers and Supplier Sites	Lists all suppliers setup in BIS with supplier sites. Pulls live data.	Information map	5/15/2017 3:11:42 PM	8/28/2017 2:08:29
AP - USAS Batch Info - Daily Outbound to USAS	Contains batches sent to USAS. Pulls live data.	Information map	5/15/2017 3:11:42 PM	8/28/2017 2:08:30
AP - Where The Money Goes - Possible Exceptions	Lists purchases that may be exceptions to the Where the Money Goes report. Pulls live data.	Information map	5/15/2017 3:11:42 PM	8/29/2017 10:05:3
AP - Where The Money Goes - Review	Lists purchases for the Where the Money Goes report. Formatted for Review. Updated Daily.	Information map	5/15/2017 3:11:42 PM	8/29/2017 10:06:2
AP - Worklists Shared	Lists worklists shared by user. Pulls live data.	Information map	5/15/2017 3:11:42 PM	8/28/2017 2:08:32
AR - Billing and Receipt History	Contains AR invoice and receipt information for AR. Pulls live data.	Information map	5/15/2017 3:11:42 PM	8/28/2017 2:08:33
AR - BIS Misc Receipts	Contains AR misc. receipt and deposit information for AR. Pulls live data.	Information map	5/15/2017 3:11:42 PM	8/29/2017 10:07:4
AR - Receivables Trade Receipts	Contains AR trade receipt information on for revenues. Updated daily.	Information map	5/15/2017 3:11:42 PM	2/28/2018 9:40:14
AR - Transaction Header Invoice Distribution Lines	Contains AR line detail transactions formatted for AR. Pulls live data.	Information map	5/15/2017 3:11:42 PM	8/28/2017 2:08:35
BD - Capital Construction	Summarized funds available for Capital Construction only. Updated daily.	Information map	5/15/2017 3:11:42 PM	8/29/2017 10:18:5
BD - Expenditure Correction Report	Contains expenditure information needed for expenditure corrections. Pulls live data.	Information map	5/15/2017 3:11:42 PM	8/29/2017 10:20:1
BD - Expense Detail	Contains detailed expenditure information for all years in BIS. Updated daily.	Information map	5/15/2017 3:11:42 PM	3/1/2018 8:06:52 A
BD - Funds Available Detail	Contains detailed funds available information for all years in BIS. Updated daily.	Information map	5/15/2017 3:11:42 PM	3/1/2018 8:07:21 A
BD - Funds Available SMART 2012	Summarized funds available for AY 12. Updated daily.	Information map	5/15/2017 3:11:42 PM	8/28/2017 2:08:41
BD - Funds Available SMART 2013	Summarized funds available for AY 13. Updated daily.	Information map	5/15/2017 3:11:42 PM	8/28/2017 2:08:43

Figure 7. An example of the naming convention used for reports in TPWD's SAS® BI system.

IMPROVE REPORTING PERFORMANCE AND MINIMIZE DISRUPTION TO END USERS

TPWD field offices are located throughout Texas, many in rural areas, which often causes significant degraded performance in the routing of data to users. Network congestion, poor local internet download speeds, an improperly-sized reporting server, and outdated computer equipment can all lead to report performance degradation, which are typically out of our control. However, because our field offices were asking for better performance on reports, our BI team explored different techniques to improve reporting performance while minimizing disruption and downtime for end users.

When first starting out, our BI team had no experience in how to best achieve different levels of optimized reporting. We decided that attending training on SAS® platform administration and SQL query design would help us develop the skill necessary to achieve our goals.

As our BI team became more knowledgeable, we leveraged our understanding developed from continuing to act in an end user role, which enabled us to translate that experience to our new role as Platform Administrators. Learning different optimization techniques applied at different levels of the end user's experience was key in achieving improved reporting performance and for minimizing user disruption and downtime.

The two main areas we targeted for improvements were the database backend and user frontend.

Database backend initiatives

Our focus for optimizing the backend of our SAS® BI database was on making sure the compute tier and storage resources were always properly maintained.

Our BI team learned how to properly optimize and create extract, transform, load (ETL) processes and schedule ETL's to make sure the majority of our ETL jobs ran during non-business hours. This allowed the compute tier to be fully available to end users during work hours.

When we began working with our SAS® 9.2 implementation, We had only had 100GB of hard drive space dedicated to our temporary WORK directory. As the use of the BI server grew, we began to encounter periodic instances where the WORK directory filled up, and caused users sessions to fail. We began using Microsoft's free Process Explorer to monitor user activity during work hours to identify specific user sessions and associated workspace folders, and manually killed any runaway or orphaned jobs, WORK folders, or WORK tables. Later we learned to use the SAS® Disk Cleanup Handler Utility to automate the process of deleting orphaned temporary SAS® files and directories on the Work directory of our SAS® server; SAS® 9.4 Maintenance Release 2 now uses the Cleanwork utility to automate this task. Although we now have a 500GB WORK directory, which alleviated many of these occurrences, we have also created an alert in SAS® Environment Manager to send e-mail notifications to Platform Administrators whenever free drive space in the WORK space reaches a predefined floor. When the alert is received, we can identify any folders on the WORK drive that are inordinately large, and contact any user associated with that folder to make sure they don't have a runaway process.

We also set up daily email alerts that provided ETL completion status reports with a timestamp on how long the process required for completion and how many data rows were included in the data table. The following code snippets form wrappers before and after the ETL project code:

Before your ETL code:

```
%MACRO START_TIMER;
%GLOBAL DATETIME_START;
%LET DATETIME_START = %SYSFUNC(TIME()) ;
%MEND START_TIMER;
%START_TIMER;
```

After your ETL code:

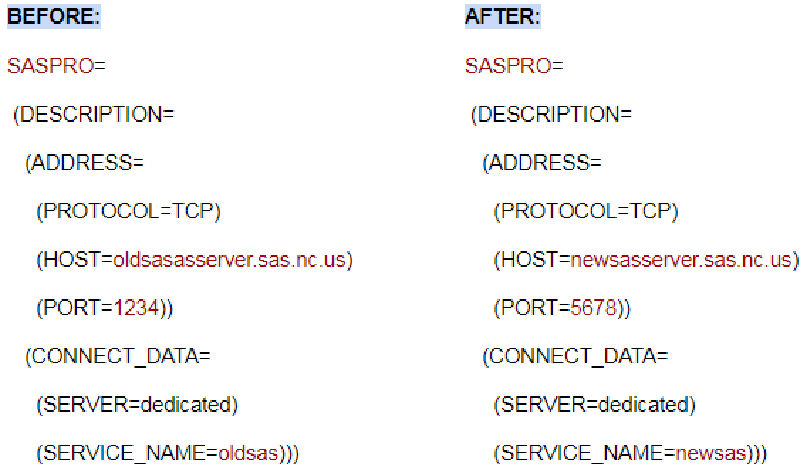
```
%MACRO END_RUN;
%PUT PROCESSING TIME:  %SYSFUNC(PUTN(%SYSEVALF(%SYSFUNC(TIME())-&DATETIME_START.),MMSS10.2))
  (MINUTES:SECONDS:MILISECONDS);
%MEND END_RUN;
%END_RUN;


PROC CONTENTS DATA=[LIBREF].[TABLE_NAME] OUT=[OUTPUT_NAME] NOPRINT;

RUN;



PROC SQL;
  CREATE TABLE WORK.[TABLE_NAME] AS
  SELECT DISTINCT t1.NOBS,
         PUT(DATEPART(t1.CRDATE),MMDDYY10.) as CRDATE,
         PUT(TIMEPART(t1.CRDATE),TIMEAMPM.) as TIME,
         CASE WHEN PUT(DATEPART(t1.CRDATE),MMDDYY10.) = PUT(TODAY(),MMDDYY10.) THEN ' ' ELSE
           '<<<<<<<<<<TABLE OUT OF DATE>>>>>>>>>>' END AS DATE_CHECK
  FROM WORK.[OUTPUT_NAME] t1;

QUIT;

DATA GRAB;
  SET WORK.[TABLE_NAME];
CALL SYMPUT('RECORD_COUNT',NOBS);
CALL SYMPUT('CREATE_DATE',CRDATE);
CALL SYMPUT('TIME',TIME);
CALL SYMPUT('DATE_CHECK', DATE_CHECK);
RUN;

FILENAME outbox EMAIL
TO="JohnDoe@email.earth.gov"
CC="JohnDoeBoss@email.earth.gov"
FROM="JaneDoe@email.earth.gov"
SUBJECT="The [Project_Name] ETL is Complete with&RECORD_COUNT Rows!"
;
data _null_;
FILE outbox;
PUT ' ';
PUT 'The Data Integration Job [Project_Name] ETL has completed.';
PUT "There were &RECORD_COUNT records.";
PUT "Table was created on &CREATE_DATE at &TIME.";
PUT "&DATE_CHECK";
put "PROCESSING TIME:  %sysfunc(putn(%sysevalf(%sysfunc(TIME())-&datetime_start.),mmss10.2))
  (minutes:seconds:milliseconds)" ;
RUN;
```

Using TNSNAMES to make Oracle server migrations less disruptive

When moving from one database instance to another database instance (in our case moving from Oracle Exadata to Oracle Supercluster) we were able to minimize downtime by applying a simple concept where the database Host and Port are changed for a defined alias using the TNSNames.ora file.

The TNSNames.ora file (located under the path “\Oracle\11g_r2\runtime\network\admin\” in our system) is a configuration file that hosts local and network service aliases, which allows connections to databases to be specified using an alias in the SAS® Metadata server.

Because TPWD’s data folder architecture is designed around the database servers, moving from one instance to another (example: Dev to Prod below) would be a simple update to the TNSNames.ora file without having to create a new folder structure in metadata, or make any other changes to the SAS® Metadata. End users would be unaware that reporting had been migrated from one instance to another, and the move would occur instantaneously as the TNSNames.ora file was overwritten. Following is an example of the simple changes that would need to be made for a specific TNSNAMES.ora entry reflecting a server migration:



BEFORE:

```
SASPRO=
 (DESCRIPTION=
   (ADDRESS=
     (PROTOCOL=TCP)
     (HOST=oldsasasserver.sas.nc.us)
     (PORT=1234))
   (CONNECT_DATA=
     (SERVER=dedicated)
     (SERVICE_NAME=oldsas)))
```

AFTER:

```
SASPRO=
 (DESCRIPTION=
   (ADDRESS=
     (PROTOCOL=TCP)
     (HOST=newsasserver.sas.nc.us)
     (PORT=5678))
   (CONNECT_DATA=
     (SERVER=dedicated)
     (SERVICE_NAME=newsas)))
```

User frontend initiatives

Our frontend initiatives focused on what reports users most consumed and how to improve the efficiency of report outputs and user processes.

As mentioned previously, our agency had recently implemented a new financial system that was built using a reporting product outside of SAS®. The new financial system had several issues, some of which resulted in inaccurate financial reporting. The accounting at TPWD is complex. Because TPWD deals with multiple funding sources, salaries, fringes, payables, capital construction can be split funded, which causes an enormous amount of budget, encumbrance, and expenditure data detail.

When reporting was created with the new financial system, very little thought was given to how the growth of data would impact a users’ ability to query for reporting. This growth in data lead to report performance degradation, aside from the other degradation issues mentioned earlier that resulted from network congestion, poor local internet download speeds, improper sizing of the reporting server, and outdated end user computer equipment.

The BI team created content from scratch using new methods which revolutionized reporting for end users at TPWD. Specific SAS® financial reporting data tables were designed using improved methods of ETL production that targeted the width and depth of data of SAS® data tables used for reports.

By separating and creating tables for each accounting year, users did not have to run report queries against a single table that contained all years. We also surveyed our end users to determine what columns of information were the most beneficial, leaving a sizeable number of columns off the final SAS® data tables.

In addition, the BI team staged the data tables within the SAS® data warehouse nightly. This minimized network traffic by keeping financial reporting all within the SAS® BI environment, as opposed to having to connect across the network to the Oracle environment containing the raw source data tables. SAS® data tables in the data warehouse included indexes on specific columns that were most commonly used in filtering.

Live reporting requires a network request outside of the SAS® BI environment, which was bottlenecking some report outputs in field offices. Instead, live reports were redesigned to be more efficient either through the use of views created via Pass-Through SQL that were optimized for Oracle, or by using SAS® data tables where their associated ETL's were scheduled to be refreshed multiple times a day.

Through the use of SAS® data tables, Stored Processes and Information Maps, reporting overall moved in a new direction at TPWD. Another advancement was the evolution of a reporting style we call SMART reporting. SMART reporting combines multiple report outputs into one report and automates several manual processes that end users formerly performed as a routine part of their jobs. This style of reporting lead to significant time efficiencies.

This paradigm shift in using SMART methods in turn led to another advancement in the automation of reconciliation. Federal and state governments, as well as private industries spend countless hours performing manual reconciliations between multiple accounting and transactional systems. Pulling unstructured data from two or more systems and trying to manually determine un-reconciled items was a process at TPWD that affected many users. By automating this process, users achieved significant time efficiencies from the process redesign which allowed them to focus more on analyzing and addressing differences.

CONCLUSION

We believe several of the philosophies and approaches that evolved during the implementation of our BI system helped us overcome inherent obstacles that commonly arise during similar efforts and were key to the success of this project. They informed our approach to developing an efficient BI architecture and an effective corresponding security model, but more importantly, led us to focus on the needs of end users, while striving to meet the requirements of Data Owners and Data Stewards. We hope some of the lessons we learned prove useful to other SAS® users who are involved in implementing a new SAS® BI system, or have inspired ideas among others who have already implemented a SAS® BI system and are seeking to improve their best practices. We also invite others to correspond with us to share their ideas for best practices that we may incorporate within our own as we continue to improve our implementation.

REFERENCES

Scheps, Swain. 2008. Business Intelligence for Dummies. Indianapolis, IN: Wiley Publishing, Inc.

RECOMMENDED READING

- *SAS® Platform Administration: Metadata Administration Course Notes*
- *SAS® Platform Administration: System Administration Course Notes*

CONTACT INFORMATION

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

John Barry Taylor
Texas Parks and Wildlife Department
John.Taylor@tpwd.texas.gov

Dan Strickland
SAS® Consulting Services
DanWStrickland@gmail.com

Alejandro Farias
Texas Parks and Wildlife Department
Alejandro.Farias@tpwd.texas.gov

Drew Turner
Texas Parks and Wildlife Department
Drew.Turner@tpwd.texas.gov