Paper 2409-2018

Developing BI Best Practices: Texas Parks and Wildlife's ETL Evolution

Drew Turner, John Taylor, Alejandro Farias, Texas Parks and Wildlife Department

ABSTRACT

The development of extract, transform, load (ETL) processes at Texas Parks and Wildlife has undergone an evolution since we first implemented a SAS® Business Intelligence (BI) system. We began constructing our data mart with subject matter experts who used SAS® Enterprise Guide® and who had a need to share that data among less-experienced staff in our user community. This required us to develop best practices for assembling a data mart architecture that maximized ease-of-use for consumers, while offering a secure reporting environment. We also developed best practice guidelines for determining whether data was best served to and from the data mart via physical SAS® tables, information maps, OLAP cubes, or stored processes. Originally, we used Microsoft Task Scheduler to automate SAS Enterprise Guide processes and developed best practices in designing process flows to run on an automated schedule. Recently, we invested in SAS® Data Integration Studio® and have developed best practices to transfer SAS Enterprise Guide process flows and schedule them within that tool. In this paper, we share the best practices we have developed in evolving ETL processes that we have found to be helpful in ensuring the success of our BI implementation.

INTRODUCTION

When TPWD implemented a new financial system in 2011 the provided reports were not able to generate adequate performance to be useful for all our field offices. Our SAS Business Intelligence (BI) team was asked to determine if we could produce reports in a better way. While they were able to successfully generate the needed reporting, they did have issues along the way. This paper serves as a case study of our experience in evolving our ETL strategy over time from utilizing the Microsoft Excel to SAS Enterprise Guide and finally into SAS Data Integration Studio. We will share with you some issues we overcame in building our data warehouse as well as our solutions to those problems.

OUR ETL EVOLUTION

As we mentioned previously, our BI team started trying to produce reports from our financial system through the only tool available at the time, Microsoft Excel. By creating an ODBC connection on a local machine into our financial database we could pull existing views from our financial system into Excel. This gave us our first real ability to determine where we were with our budgets. We saved these workbooks onto our agency-wide shared network drive which allowed end users access to this data. As we received requests for more complex reports we were given SAS® Enterprise Guide® (EG).

ENTERPRISE GUIDE

Using SAS® Enterprise Guide® (EG) with a local ODBC connection allowed us to do joins that we were unable to do through Excel alone. Our first ETL was pulled thru EG and output as Excel onto a shared network drive. The graphic user interface and a command line code node EG provides allowed us to quickly learn how SQL code was generated and provided instant coding feedback.

While learning SQL coding we utilized not only EG but also the free tool SQL Developer provided by Oracle. This allowed us to test our SQL before implementing it within EG. Since our financial system runs on Oracle this tool has additional features which speeds our ability to generate and troubleshoot SQL code.

SAS BI SERVER

Once we had proven that pulling data through EG was a solution to our reporting problem we were introduced to the SAS BI platform. We realized that the SAS BI platform would allow us to scale up our reporting in a more secure way than sharing Excel workbooks as well as ensuring that each report was displaying the answer for the same question. We created a SAS/Access for OLEDB® connection on our SAS BI Server and began staging data on the SAS server instead of exporting directly to Excel. We began providing the SAS Add-in for Microsoft Office in Excel to our users to query our staged data. Over time we learned of the other tools available in the SAS BI suite including Information Maps, Stored Processes and OLAP Cubes. We learned the strengths and weaknesses for each tool and implemented them as accordable to enable access to our data warehouse. Around this time, we upgraded our connection from the SAS BI server from an OLEDB connection to an SAS/Access for Oracle® connection which decreased our ETL jobs' runtimes significantly.

We were still using Microsoft Task Scheduler to schedule Enterprise Guide ETL projects on our desktops as we didn't have access to schedule ETL jobs on our SAS server at the time. We encountered many issues with this approach with the most troublesome being updating passwords, unexpected power outages and unexpected reboots for Windows patches that caused the scheduled jobs to fail. After sufficient SAS platform administration training, we were granted access to our SAS server tier. At this time, we also upgraded from a single tier on SAS 9.3 to a SAS 9.4 split-tier system with a separate server for the metadata, compute and middle tier servers. We also purchased SAS Data Management Studio® and Data Integration Studio. Due to the issues previously listed with our scheduled EG jobs, we decided to migrate our EG jobs to Data Integration Studio.

SAS DATA INTEGRATION STUDIO

After receiving SAS training on Data Integration Studio® (DIS), we began migrating jobs that were previously deployed and scheduled through EG to DIS. This allowed us to schedule these jobs on the SAS server where power failures and unexpected reboots are uncommon. To migrate from EG to DIS we evaluated the tools available to us in DIS and settled on using mostly user written code nodes. Since we were using query builder nodes in Enterprise Guide, we could easily export the generated SQL code from the query builder nodes and create user written code nodes in DIS. We continue to use this setup today and it has served us well with little down time and high reliability for our reports to our end users.

LESSONS LEARNED

Over the course of our ETL evolution we learned some lessons we would like to share, including tips on segmenting datasets, adding value added columns to tables, indexing and filters, compress system option usage, SQL Developer usage, job status emails, content naming convention, scheduling jobs and SAS Data Integration Studio usage.

SEGMENTING DATASETS

When we first started to produce reports from our financial system we were querying a flat file that contained over 200 columns and in the first year alone grew to over 10 million rows. We met with our end users and determined which columns they needed for the majority of their work and created an ETL that pulled only those columns for the current fiscal year. By segmenting the data by fiscal year, we can also decrease how much data needs to be filtered to return a result query.

VALUE ADDED COLUMNS

At the same time, we spoke to end users about what columns they could choose to segment our data, and we asked what columns could we add to make them more efficient. Those columns we added to make staff more efficient include a grouping column specifically for a division, a concatenation of two existing columns, and converting a flag from a single digit to a more understandable term, 'P' to 'Posted' for example. By using case statements, we can allow users to provide us with a list of conditions they want satisfied before a certain value is displayed. An example computed column is shown in Figure 1 below which assigns a division based upon the first two letters of a project name.

```
CASE
          WHEN SUBSTR (PROJECT,1,2)='AR' THEN 'Administrative Resources'
          WHEN SUBSTR (PROJECT,1,2)='CC' THEN 'Capital Construction'
          WHEN SUBSTR (PROJECT,1,2)='CF' THEN 'Coastal Fisheries '
          WHEN SUBSTR (PROJECT,1,2)='CL' THEN 'Capital Land Acquisition'
```

Figure 1. Example computed column code

INDEXING DATASETS

By segmenting our data into smaller datasets we are also able to index the data according to how our users pull it. There are several strategies to adding indexes to datasets but we have found that indexing columns that are used in filters in our reports has the best results for us. This index is a pre-summarized collection of what row on the dataset that value is on. This collection allows SAS to choose those rows faster than looking though all the rows in the dataset.

COMPRESS SYSTEM OPTION

In talking to a SAS consultant regarding best practices for administering our BI system we learned about the compress system option. This option compresses output SAS datasets when they are written to our BI server. This greatly reduces the amount of storage space needed for our ETL tables. This option can be assigned at the system level, the library level or in an option statement. For our agency we set this option at the system level by including '- compress yes' in our sas9v_usermod.cfg file located at \..\sasconfig\Lev1\SASAPP\ .

JOB STATUS EMAILS

Running multiple jobs, each day can make verifying they ran successfully challenging. We have made a practice of adding email code to the end of every ETL job that we create and deploy. I wrote a paper for the 2012 SCSUG conference that contains details on implementing this process (Turner 2012). By receiving emails with the line count and run times for our jobs we can not only verify that an ETL ran but this also allows us to gauge how long each job takes to run. We can adjust our start times for our scheduled jobs to ensure we don't run too many of our jobs at once and run out of server resources. We can also send emails when a job has an error which needs our attention. This makes managing and running over 50 daily jobs manageable for our team.

SQL DEVELOPER

We have found that the when programming in SQL to access our Oracle databases, Oracle SQL Developer offers worthwhile advantages to create the most efficient code development process. We utilized the SQL Developer tool provided by Oracle to develop SQL coding that is utilized in ETL's and reports. By using SQL Developer, we are able quickly troubleshoot code as the error messages in SQL Developer provide a line number identifying the problem line. It also has a code formatter that enhances readability of SQL code. The ability to run explain plans allows us to tweak our queries to utilize fewer resources and run faster ETL's. An example SQL Developer window is shown below in figure 2.

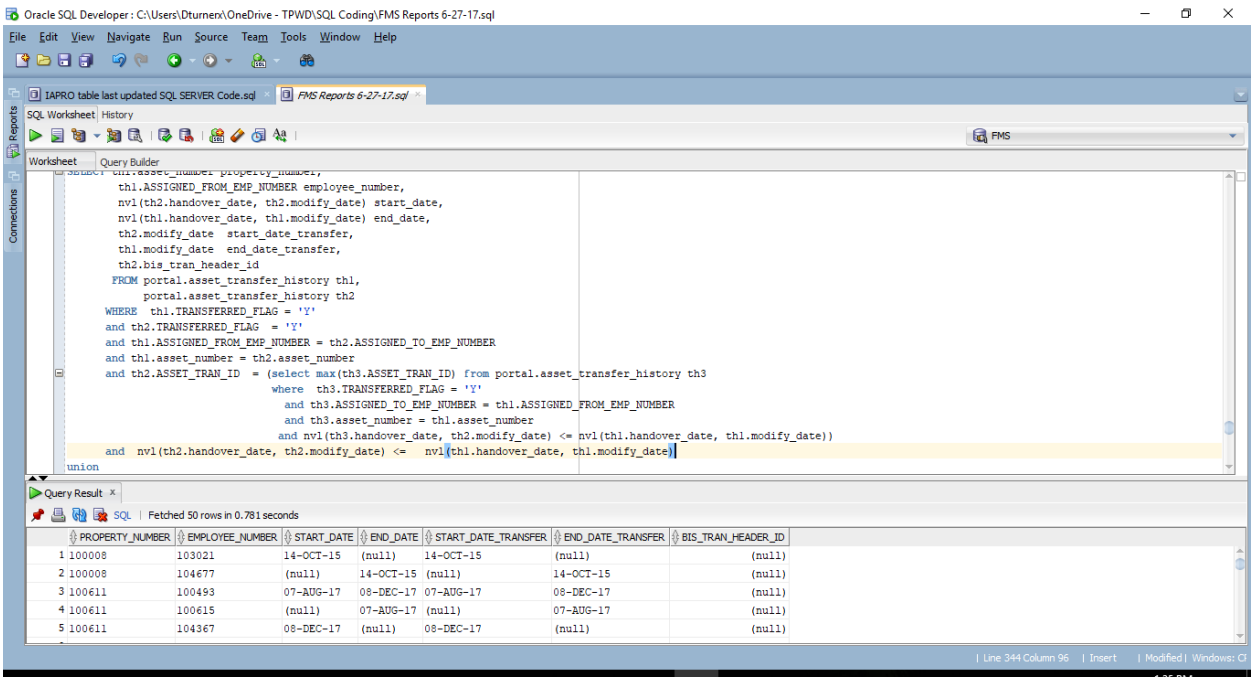


Figure 2. SQL Developer code and result

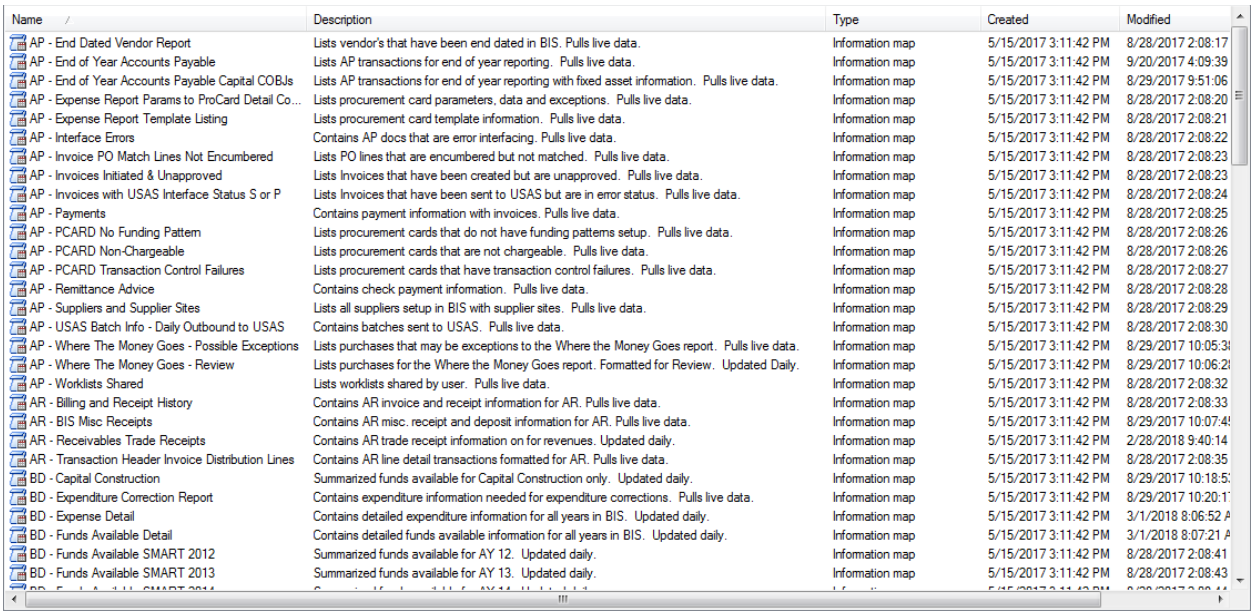
ESTABLISH A NAMING CONVENTION FOR CONTENT

After building and distributing a number of reports to our end users, we committed to moving all financial system reports onto SAS and shutting down our existing non-SAS reporting tool. We realized that with the additional reports it would be confusing for users to have to sort through dozens of new reports we were about to create. We decided that we needed to rename all our reports following a naming convention that organized the reports by content area. We created a report catalog which classified reports by group (below in Table 1), then give a specific name with a short description of what the report achieves and if the underlying data is live or updated daily. Catalog 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. Report group catalog

Users can quickly identify what they are looking for and if the report outputs suit their specific needs. Each aspect of the catalog (group, name, and output description) is maintainable and easy-to-comprehend for your users. An example of our report naming convention is shown below in Figure 3.



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 3. Screenshot of report naming convention

SCHEDULING

As we stated before we first utilized the Microsoft Task Scheduler to schedule Enterprise Guide ETL projects on our desktops as we didn't have access to schedule ETL jobs on our SAS server at the time. We encountered many issues with this approach with the most troublesome being updating passwords, unexpected power outages and unexpected reboots for Windows patches that caused the scheduled jobs to fail. After sufficient SAS platform administration training, we were granted access to our SAS server tier. At this time, we also upgraded from a single tier on SAS 9.3 to a SAS 9.4 split-tier system with a separate server for the metadata, compute and middle tier servers. We also purchased SAS Data Management Studio and Data Integration Studio. Due to the issues previously listed, we decided to migrate our EG jobs to Data Integration Studio which allowed us to schedule ETL jobs on the SAS server.

DATA INTEGRATION STUDIO

After receiving SAS training on Data Integration Studio (DIS), we began migrating jobs that were previously deployed and scheduled through EG to DIS. We evaluated the tools available to us in DIS and settled on using mostly user written code nodes. Since we were using query builder nodes in Enterprise Guide, we could easily export the generated SQL code from the query builder nodes and recreate them in DIS. This ability saved us time and effort in migrating to DIS. An example EG and DIS job can be found below in Figures 3 and 4.

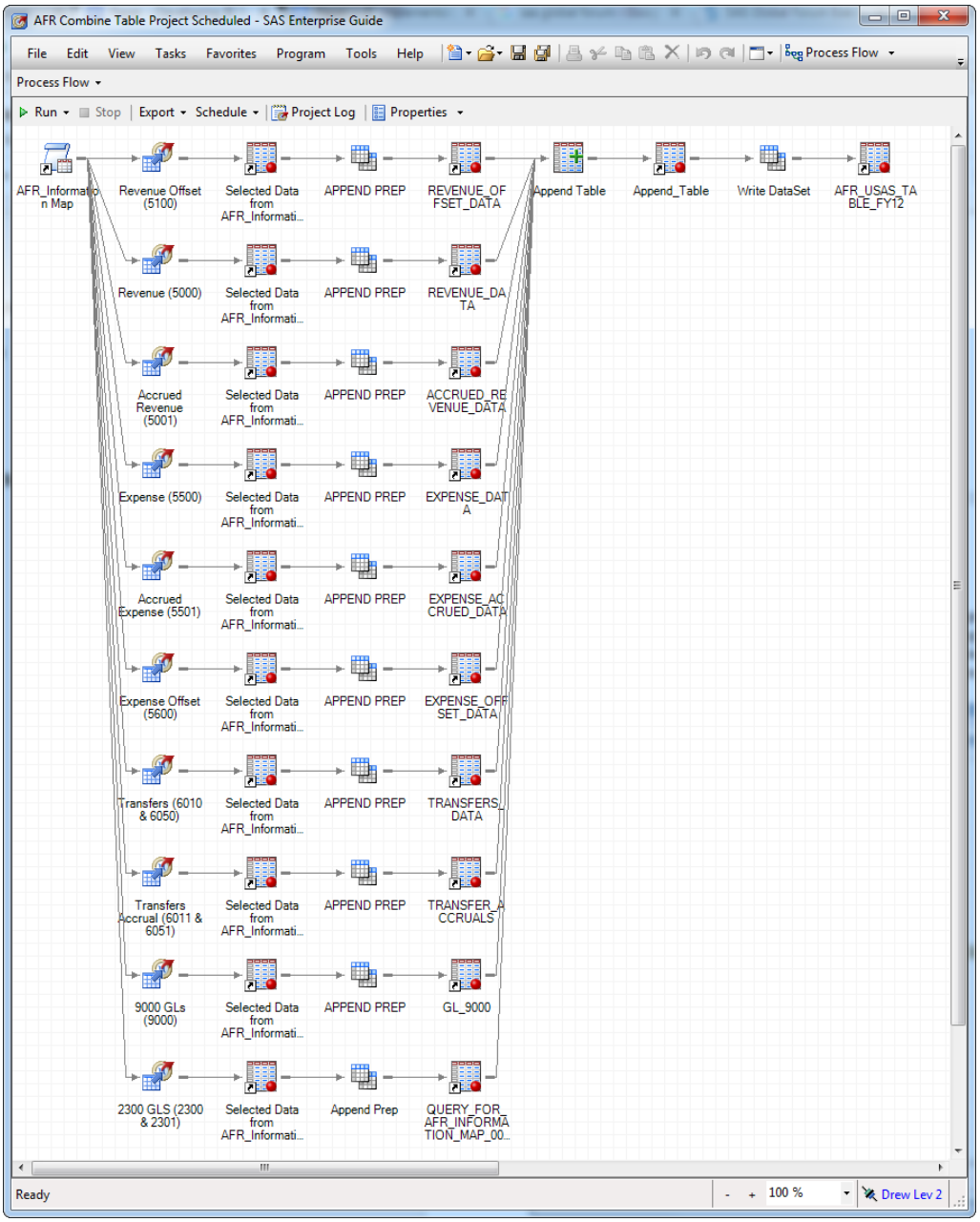


Figure 3. Screenshot of example EG job

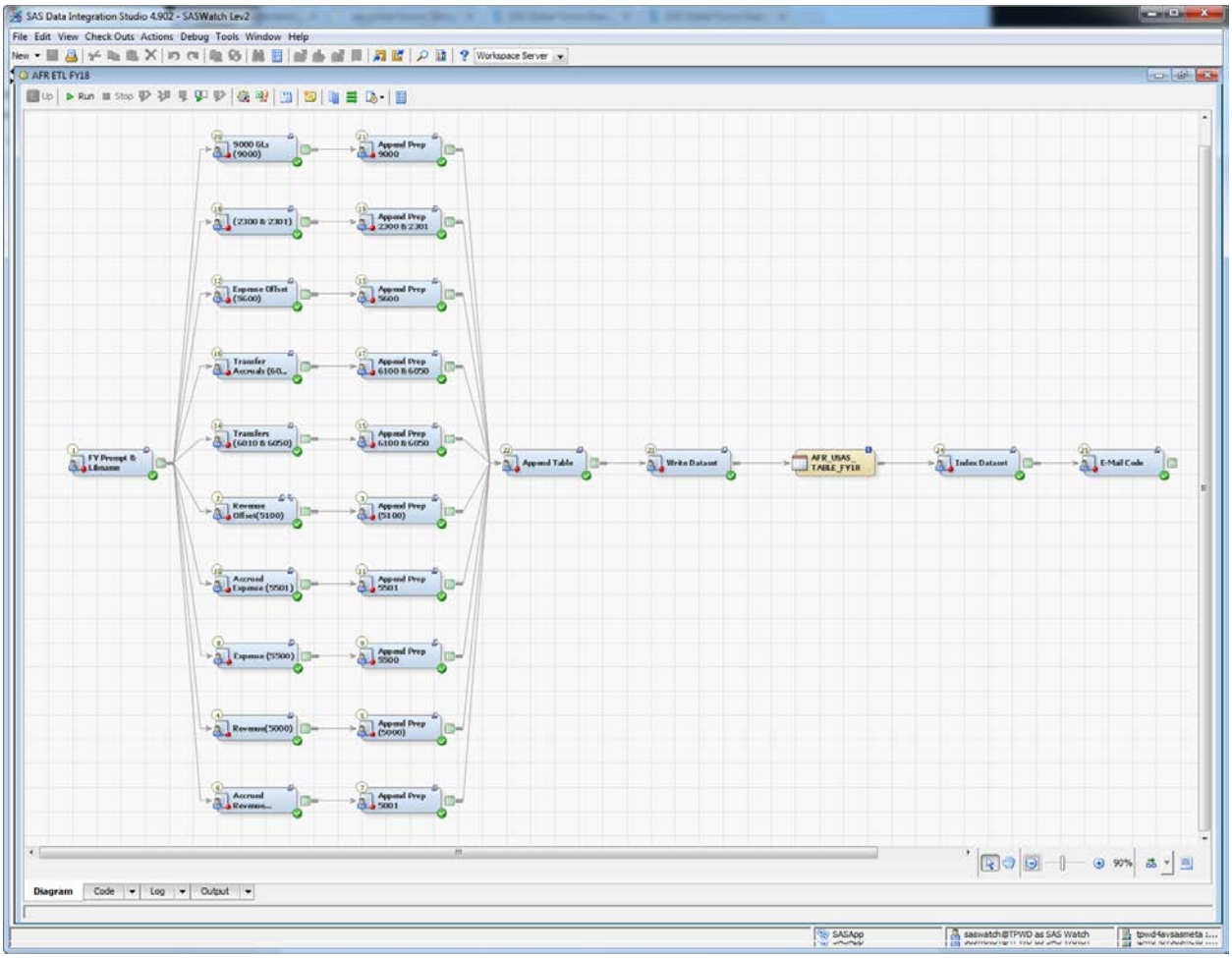


Figure 4. Screenshot of example DIS job

We learned that when using DIS, the username and password that you use to schedule the job is used to run the job on the server. After having jobs fail due to our personal accounts being locked out, we created a dedicated domain user account specifically to schedule and run jobs on the SAS server. This domain user account has a non-expiring password that simplifies maintenance and job management. This user account is also in the metadata with the needed read and write permissions. This process has mostly eliminated jobs failing due to a locked personal account.

Once the job is deployed through DIS (Figure 5), an administrator creates a workflow through SAS Management Console, which creates the scheduled job on the servers' Windows Task Manager.

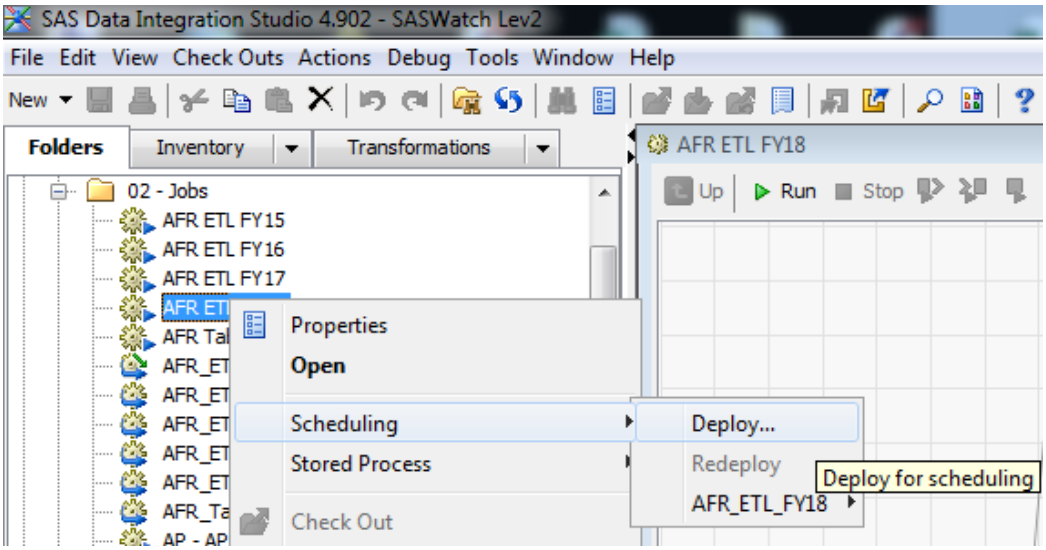


Figure 5. Screenshot of deploying a job through DIS

As you can see below in Figure 6, we have a number of DIS flows that control our ETL jobs. The naming convention that was originally developed for end user reports also served in naming DIS jobs, allowing us to keep the naming convention consistent between reports provided to end users and DIS jobs created by our BI Team. To reduce confusion, we create a new flow for each job we deploy.

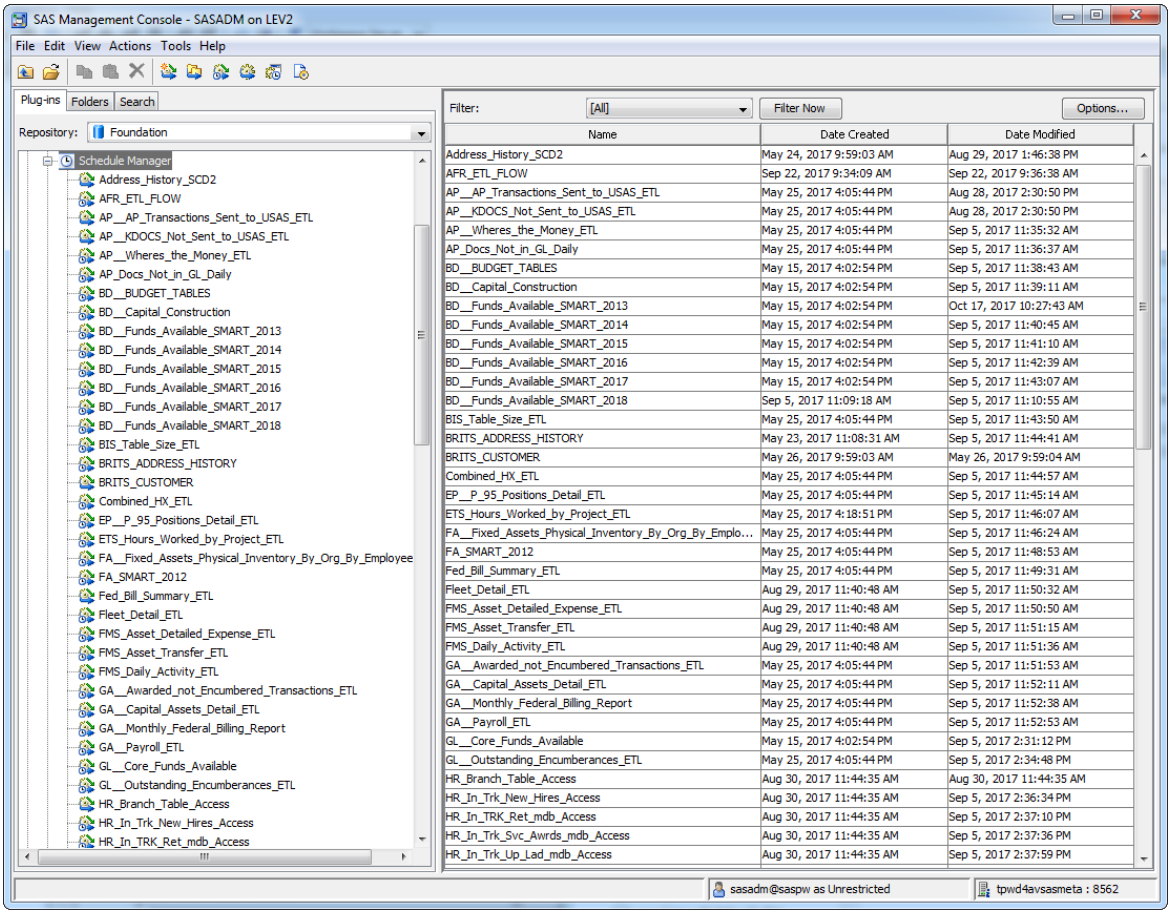


Figure 6. Screenshot of scheduled jobs through management console

We do have exceptions to this rule, as job flows do have the ability to run multiple jobs in a user defined order. An example of this is seen below in Figure 7. This example requires a table to be created before additional manipulation and logic can create our final output table. Since we use the same prerequisite table to produce multiple fiscal-year-specific tables, this arrangement reduces duplicative work each year.

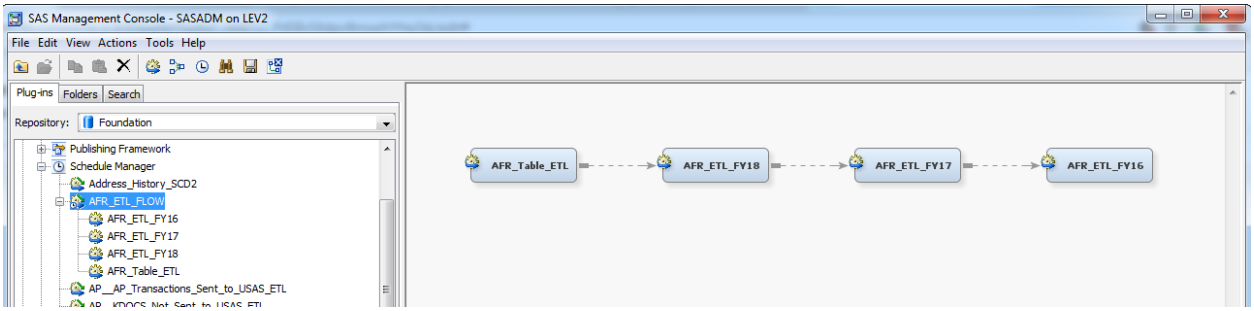


Figure 7. Screenshot of multi-job flow in management console

Once the job flow is created, it is deployed to the SAS server through management console. We use the built in Windows scheduling server to schedule the job on the SAS server. Remoting into the SAS compute tier server, you can see the job has been created with the dedicated user account we used to deploy the DIS job (saswatch) as shown below (Figure 8). With the dedicated user account, which has a non-expiring password, we are done with this DIS job until the next fiscal year.

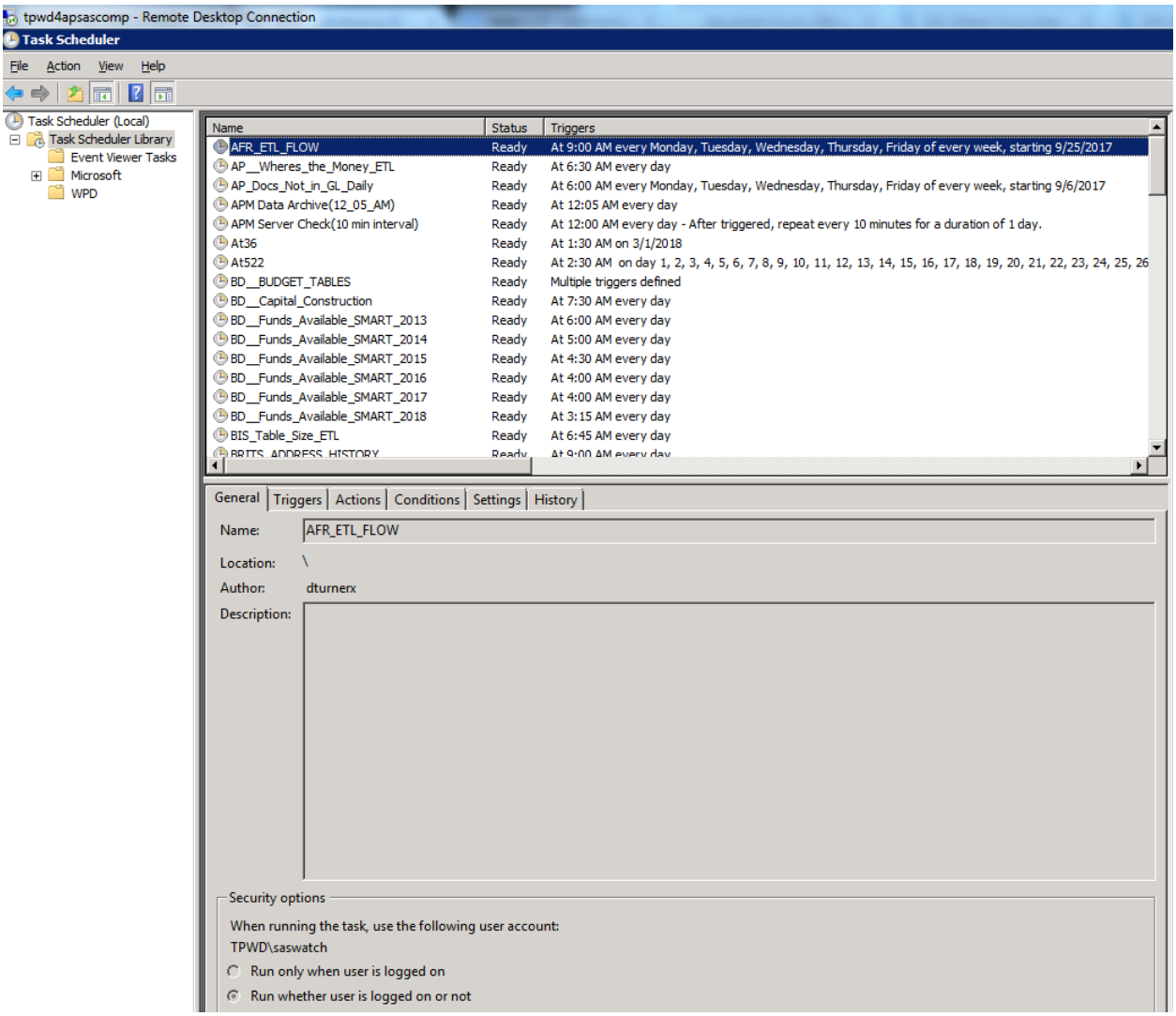


Figure 8. Screenshot of scheduled jobs in Microsoft Task Scheduler on SAS Compute tier

CONCLUSION

As you can see from our experience above, our methods have evolved over time as we gained experience. We started developing reports using an ODBC connection through Excel and progressed to using Enterprise Guide and finally to Data Integration Studio for our ETL jobs. We first used the Microsoft task scheduler on our local desktops but have moved to using DIS to schedule our jobs on our Server tier with Microsoft Task Scheduler. We hope our experiences can assist others in developing a successful ETL strategy.

REFERENCES

When was my data last updated? Turner, Drew. November 6, 2012. "When was my data last updated?." Proceedings of the SAS South Central Users Group Meeting 2012, Houston, TX. Available at http://www.scsug.org/wp-content/uploads/2012/11/When-was-my-data-last-updated-by-Drew-Turner-SCSUG-2012.pdf.

RECOMMENDED READING

- *Base SAS® Procedures Guide*
- *SAS® For Dummies®*
- *SAS® Data Integration Studio User's Guide*
- *Compress System Option:*
  http://support.sas.com/documentation/cdl/en/lrdict/64316/HTML/default/viewer.htm#a000202890.htm

CONTACT INFORMATION

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

Drew Turner
Texas Parks and Wildlife Department
(512) 389-8246
Drew.Turner@tpwd.texas.gov

Dr. John Taylor
Texas Parks and Wildlife Department
(512) 389-4338
John.Taylor@tpwd.texas.gov

Alejandro Farias
Texas Parks and Wildlife Department
(512) 389-8154
Alejandro.Farias@tpwd.texas.gov