Paper 136-2008

Data Integration with SAS® Intelligence Storage: How a Business Division in a Fortune 50 Company Uses SAS® Scalable Performance Data Server to Make Better Decisions

Stephen Scott, Global Software Development Services Inc.,
Justin Eisenzimmer, Brian Herzog, Target Corporation, Minneapolis, MN

ABSTRACT

Our business division integrates our data sources with SAS® into one of the world's largest SAS® Intelligence Storage systems. Our largest component is SAS® SPD Server data libraries with 25 TB (terabytes) of storage where we have analytical control of our data. We also have SAS® OLAP Server cubes at 500 GB and SAS® data sets with 600 GB plus a 2-terabyte area for Working Storage.

The corporate SAS® platform is a 100+ CPU UNIX system dedicated to SAS processing of which our business group's domain is 24 CPUs. The corporate SAS physical storage is in a dedicated SAN of 100+ TB, and our portion is 25 TB.

The SAS Intelligence Storage includes a production SPD Server area where data sources are integrated via a job scheduler running over 800 SAS jobs on a daily and weekly basis. The next level of integration is via business power users with SAS® Enterprise Guide®, SAS® Data Integration Studio, and SAS® OLAP Cube Studio which they use to integrate data and store it in their own shared SPD server libraries and OLAP cubes for further analysis. This gives the business the creative freedom to respond to changing business requirements.

In this paper, we will:
• describe the overall architecture of the environment
• discuss the production-level data integration methods
• show how we integrate data sources from the business side
• demonstrate how we make better business decision via data integration.

INTRODUCTION

This paper is concentrating on the building and usage of a 25+ terabyte SAS® Scalable Performance Data Server area for the Strategic Pricing business area and how and why we architected it from a technical and business viewpoint. That is why the authors and presenters are from that partnership between IT and the Business that is necessary for success.

We will cover the Business owner and analyst's perspective and the IT development perspective.

SAS Business Analysts and Admin Perspective:

SAS DATA WAREHOUSE

ENVIRONMENT

Our data warehouse is the basis for the analytics analysts perform to make better business decisions every day. We currently have one of the largest SAS SPD Server environments in the world. The environment consists of a Production, Open, Share and Development library as well as a user work area. The warehouse is populated and refreshed though a weekly SAS ETL process. This process consists of over 140 SAS jobs scheduled on a CA-7 job scheduler server (Figure 1) and 800 SAS jobs scheduled on a BMC-Control M scheduler on UNIX. These jobs extract data from our transaction applications. These data extracts are then augmented from other data sources. Calculations are performed to create applicable metrics for the business units that exploit the warehouse. Information is restated at several levels of varying detail for optimization of reporting and analysis.

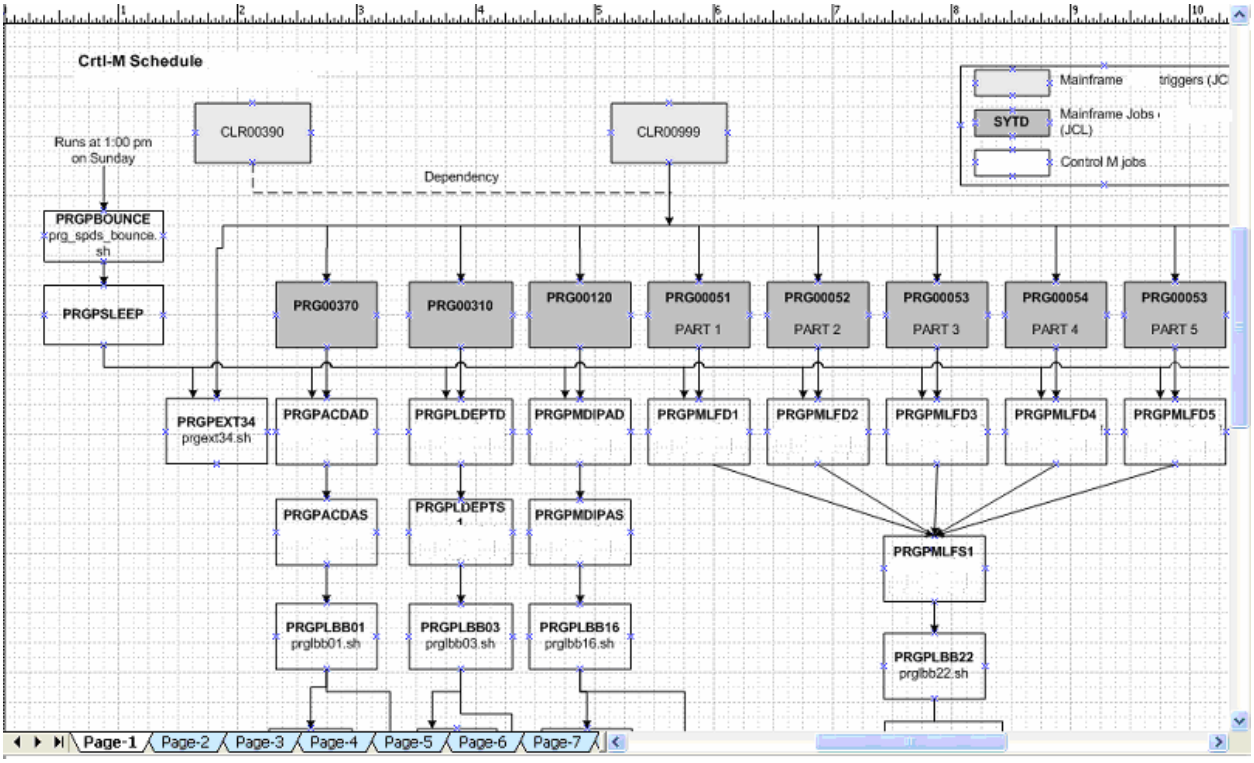


Figure 1: SAS Production ETL job process flow

Security is setup to ensure our production library is only updated through our scheduled batch jobs. The Open, Share and Development libraries are different. Analysts are able to write query results not only to their work area, but also to Open and Share. This allows the business units to share potentially millions of rows of data with one another for further analysis. The ability of sharing this data allows units to make business decisions on relevant data in a speedy fashion.

The Open and Share environments also allow the creation of additional ETL processes at a moments notice. If we need to access our mainframe and begin to pull and summarize certain information we currently do not store in our warehouse, our Business Intelligence team is able to put together a process immediately and begin to realize the benefits of the data. The alternative would be to involve IT and create a project to have the same thing done, which would be slower and more expensive. Being able to create a simple ETL process and schedule it using DIS allows us to meet our business needs quicker resulting in bottom-line savings.

BENEFITS

There are many benefits of our data warehouse and the use of SAS® Scalable Performance Data Server.

Through the creation of our warehouse, we are able to improve the response time of our transactional systems since the bulk of reporting is run from the data warehouse.

The warehouse reporting is optimized for the warehouse whereas the data in the operational systems is not conducive to reporting at the level business units require. By using SAS Scalable Performance Data Server we also realize improved speed of processing large amounts of data by partitioning the data across multiple disks and I/O channels. See Figure 2. Our largest table contains 2 billion rows of data. Thru the use of SPDS, to query and process large amounts of data in a reasonable amount of time is possible and something our analysts do on a daily basis. For example, one of the most commonly used SPD Server tables contains over 800 million rows of data. Enterprise Guide is able to access and return subsets of data in minutes or even seconds. This example of a simple query of this table returned nearly 1 million rows in 8 seconds (Figure 3). This is due to indexes and the construct of this SPD Server table.

Via the use of a SAS warehouse, we are able to take several transactional applications and bring data together into one place for analysis and reporting purposes. Reporting and analysis of data from multiple systems is possible from the data warehouse, this same reporting is not available at the application level. Reports are written and analysis is done from the warehouse using Enterprise Guide. Analysts can perform these activities in an on-demand fashion. To write these same reports from the transactional applications would take much more time and would be much more expensive to produce.

Trend reporting is available in the data warehouse due to the accumulation of years of data. This data is not kept in the applications that feed the warehouse. Critical forecasting and trend functions are able to be done quickly and are more reliable due to the existence of this information.

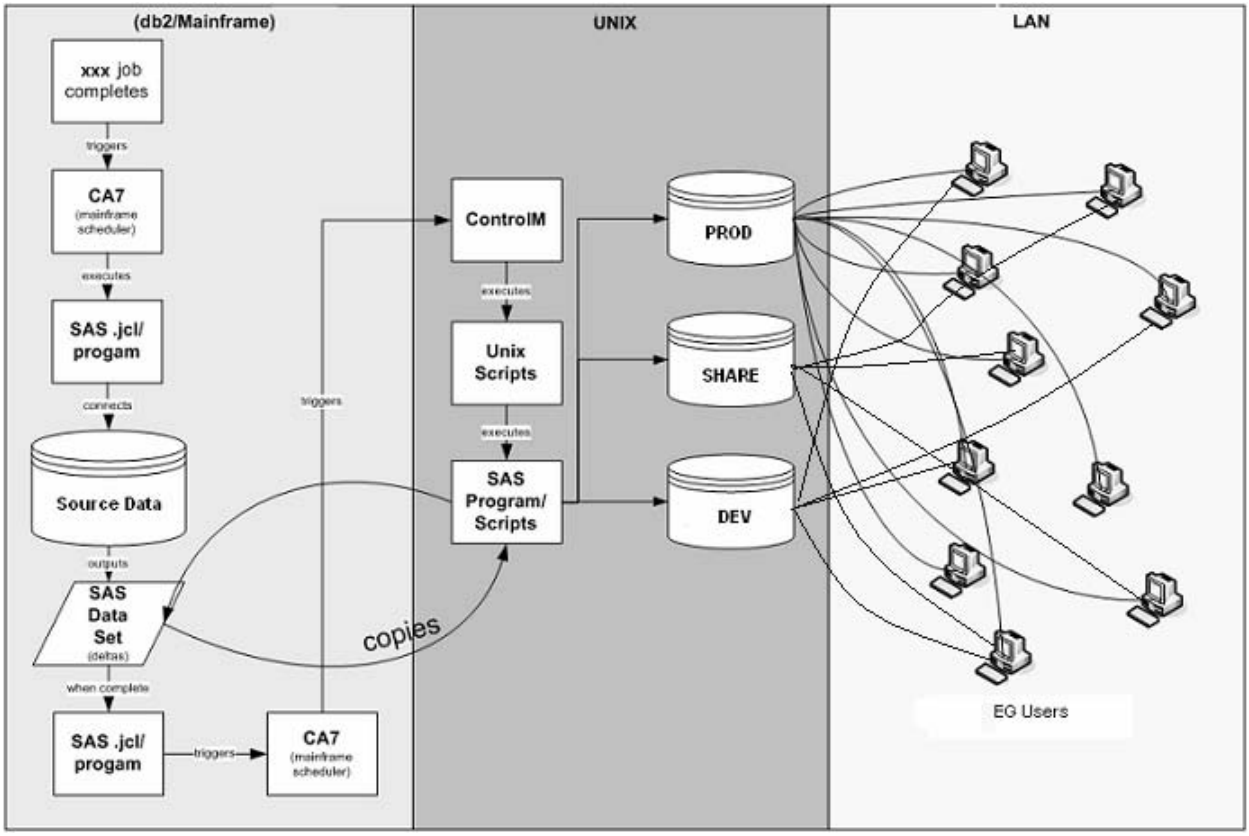


Figure 2: SAS ETL Architecture for SPD Server areas and EG Connectivity

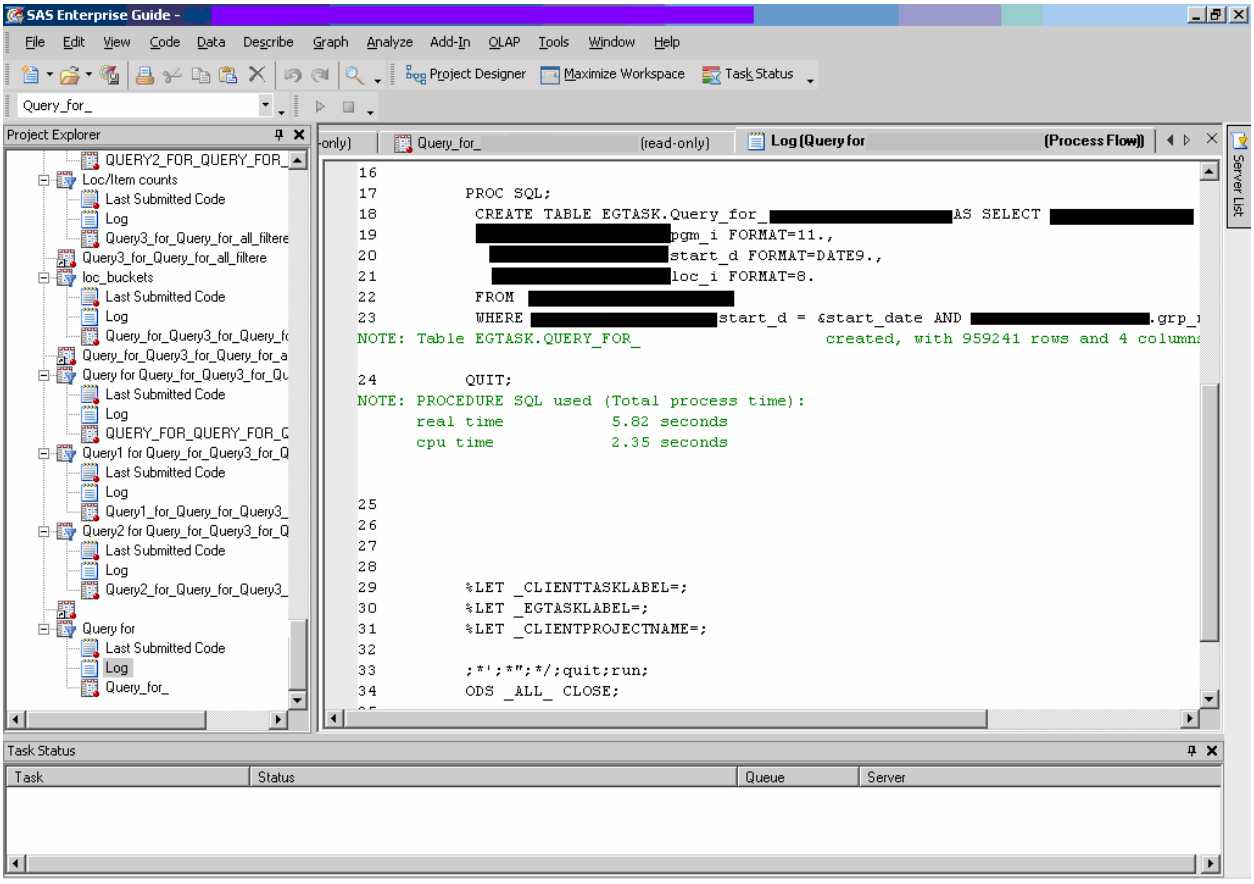


Figure 3: EG query log selecting 1 million from 800 million row table (approx.)

SAS TOOLS

The analysts use SAS Enterprise Guide as their primary tool to explore the SAS data warehouse. The Business Intelligence team however uses a variety of SAS products to access, report on and augment the data warehouse.

SAS® ENTERPRISE GUIDE

Analysts use SAS Enterprise Guide to build their own ad-hoc queries and run pre-built projects against the data warehouse. It is used primarily to access our warehouse. Enterprise Guide is also used to access other data sources: DB2® tables, Oracle® databases, Access®, Excel®, text and csv files to name a few. These other data sources are pulled in to augment the data in the warehouse when needed. The end result is access to data that is easily and quickly obtained to provide solutions to business questions.

PRE-BUILT PROJECTS

The Business Intelligence team builds projects to answer frequently asked business questions. The use of process flows and parameterized queries is implemented to allow for quick and easy user input and launch of reporting and analysis. The user simply right-clicks on the process flow to run, and necessary pop-ups prompt them for parameters for the process.

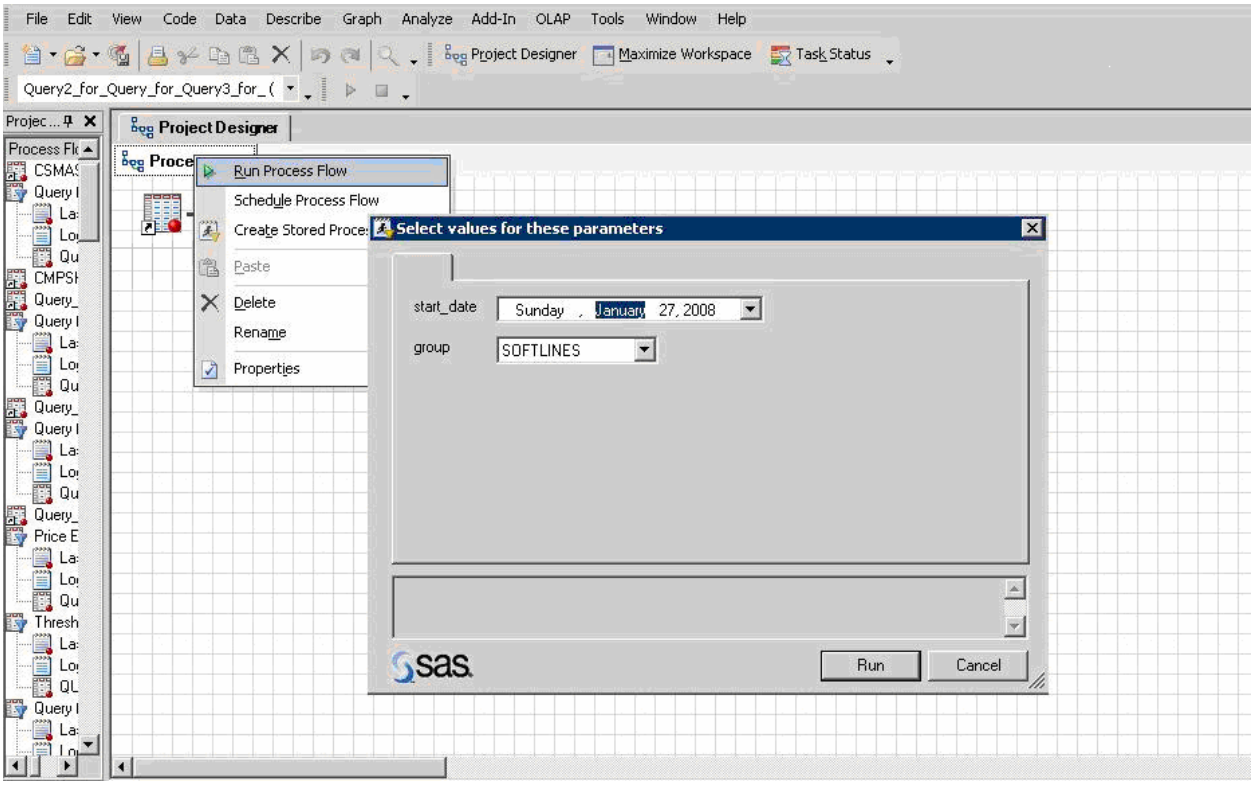


Figure 4: EG query prompt design for business analyst

The business analyst does not have to deal with the joins and calculations of more complex projects. Access to our DB2 environment is one of the most powerful aspects of SAS Enterprise Guide for our business analysts. This unlocks the power of access to many of our enterprise application database tables for simple querying. By simply including a macro in our EG code, we are able to access our DB2 tables.

Login validation:

```
SIGNON remote here NOSCRIPT
USER = xxx
PW = 'xxxxx';
```

Mainframe environment variables:

```
RSUBMIT REMOTE = xxxxx
WAIT=YES       /* RUN SYNCRONOUS/SERIAL */
PERSIST=NO     /* AUTOMATICALLY SIGNOFF AT ENDRSUBMIT */;

LIBNAME ENVIRONMENT DB2 AUTHID=XXX SSID=XXXX;
LIBNAME mffile 'mainframe filename here'; RUN;
```

Once we include the code to connect to the mainframe, we are able to upload our tables from our SAS environment to the mainframe. We can run queries off of the uploaded files, joining them to DB2 tables. Finally, we can download tables from the mainframe back into our SAS environment for further processing.

```
%include '~/Macros/MainFrameIDPW_Macro.sas';
%include '~/Macros/MainFrameLibrary_Macro.sas';
run;
proc upload data=mytable out=mffile.mytable;
run;

proc sql;
create table mffile.mynewtable
as SELECT
N.COL1,
```

```
DB2.COL2,
DB2.COL3
FROM
(db2prod.DB2TABLE DB2, mffile.mytable M
WHERE
M.COL1 = DB2.COL1;
quit;

proc download data=mffile.mynewtable out=work.mynewtable;
run;
endrsubmit;
quit;
```

AD-HOC REPORTING

The primary focus of training is to educate the analysts on the warehouse. Analysts are introduced to where their data is and what it consists of. Once they know what is available to them, they are eager to learn a tool to enable them to get at the data as they use this information to answer business questions more efficiently and confidently. A minimal amount of training on SAS® Enterprise Guide is required to get analysts off and running. Step one is to know in which library the data resides. Step two is to know which table(s) they need to pull information from based on the type of data and at what level it resides. After that, they just use Enterprise Guide to pull and format their information. Every day our analysts use Enterprise Guide to gain knowledge of their business thru the SAS data warehouse. Enterprise Guide offers analysts summary tables, list data, distribution analysis and many other tools to perform reporting and analysis of their information. The analysts have access to reliable data and use EG for fast analysis on millions of rows of data at various levels of detail. The tables are easily accessed through EG via a tree view of available libraries and tables (Figure 5). This is all put together to provide accurate insight for making smart business decisions in a timely manner.

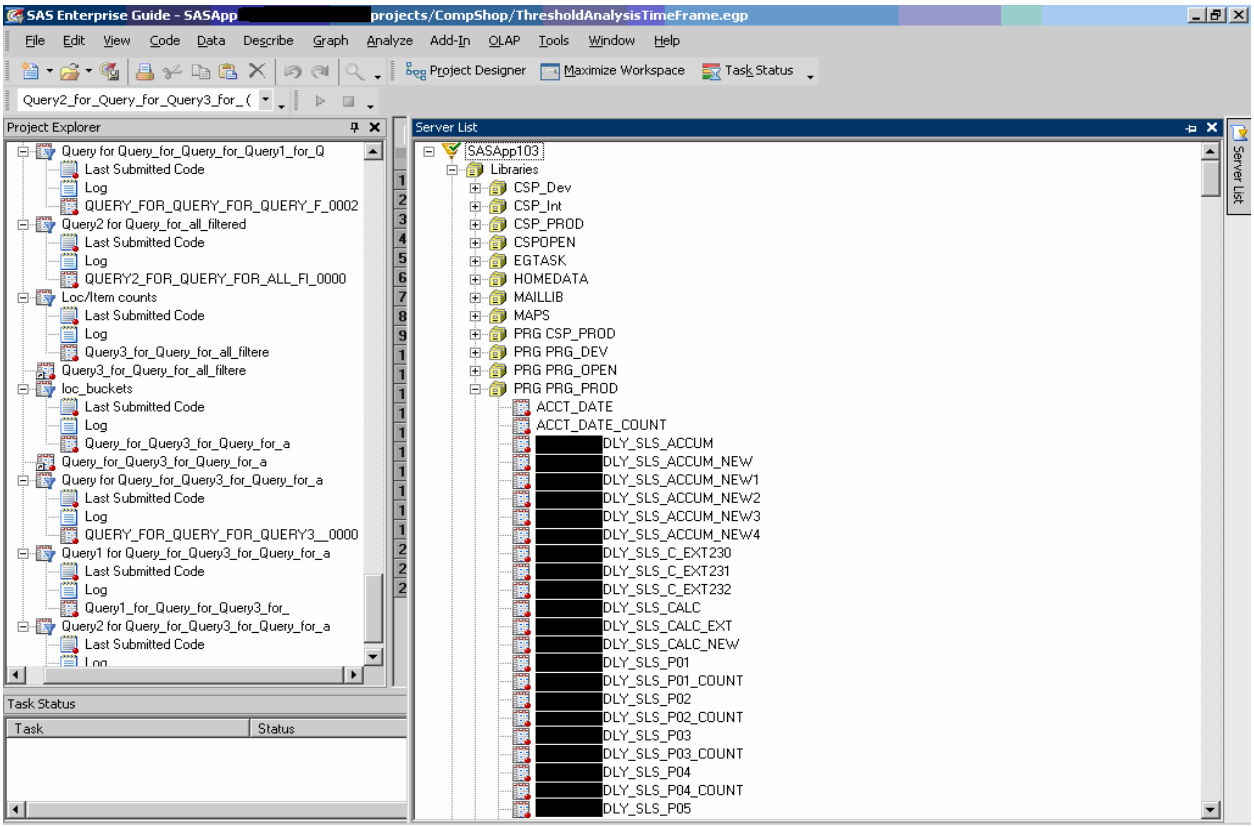


Figure 5: SAS EG library list for SAS SPD Server libraries

SAS OLAP CUBES

One of the biggest benefits available to our SAS Group is the ability to create multi-layered SAS OLAP Cubes through SAS Data Integration Studio (and from SAS SPD Server tables). Looking at our data structured as SAS OLAP Cubes allows for our users to fly through their information to different leveled hierarchies with ease and speed. This method of viewing data can save users good amounts of handfuls of time by eliminating the necessity of accessing two, sometimes three, two-dimensional data sources (such as tables).

Similar to composing a query in SAS Enterprise Guide, as covered earlier, constructing a cube can be quite simple after the process has been mastered by a couple initial composites. Like SAS EG, SAS Data Integration Studio uses friendly point-and-click pop-up windows to guide users through their data cube modeling.

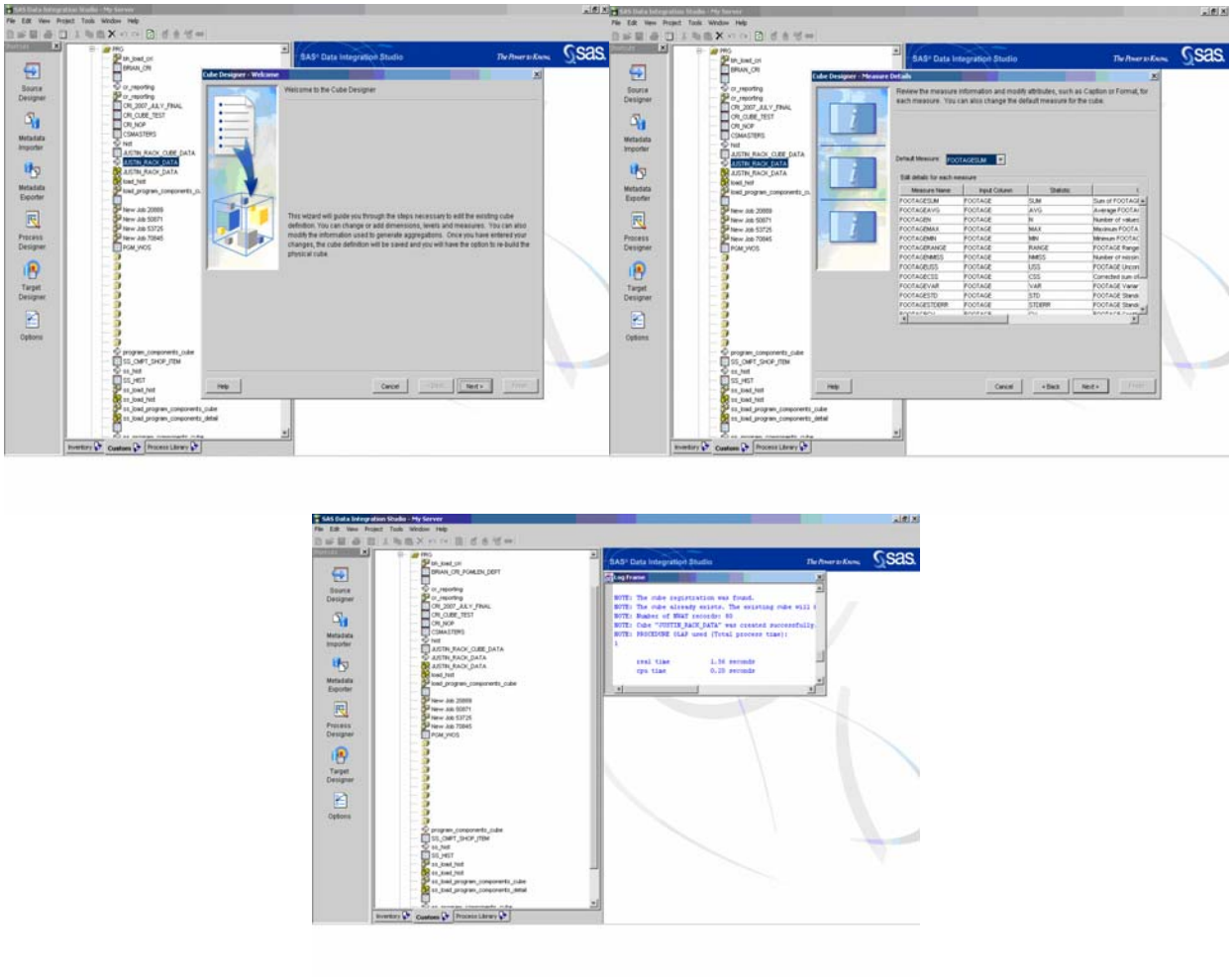


Figure 6: SAS® OLAP Cube Building via SAS® DI Studio

All the wizard's windows need telling is its data source, dimensional definitions (the hierarchies), selected members and their computed statistical functions (called measures), and a spot to be housed; the single, most difficult hurdle to overcome with building cubes is deciding which data wouldn't be looked at beneficially through SAS OLAP Cubes—as most data has great benefit in being looked at summarized, granular, and all in between.

What makes the process of structuring data in cube form one of the most choice methods of viewing data—save for the hierarchy drill-thru feature previously mentioned—is its user-friendly compatibility with SAS Enterprise Guide.

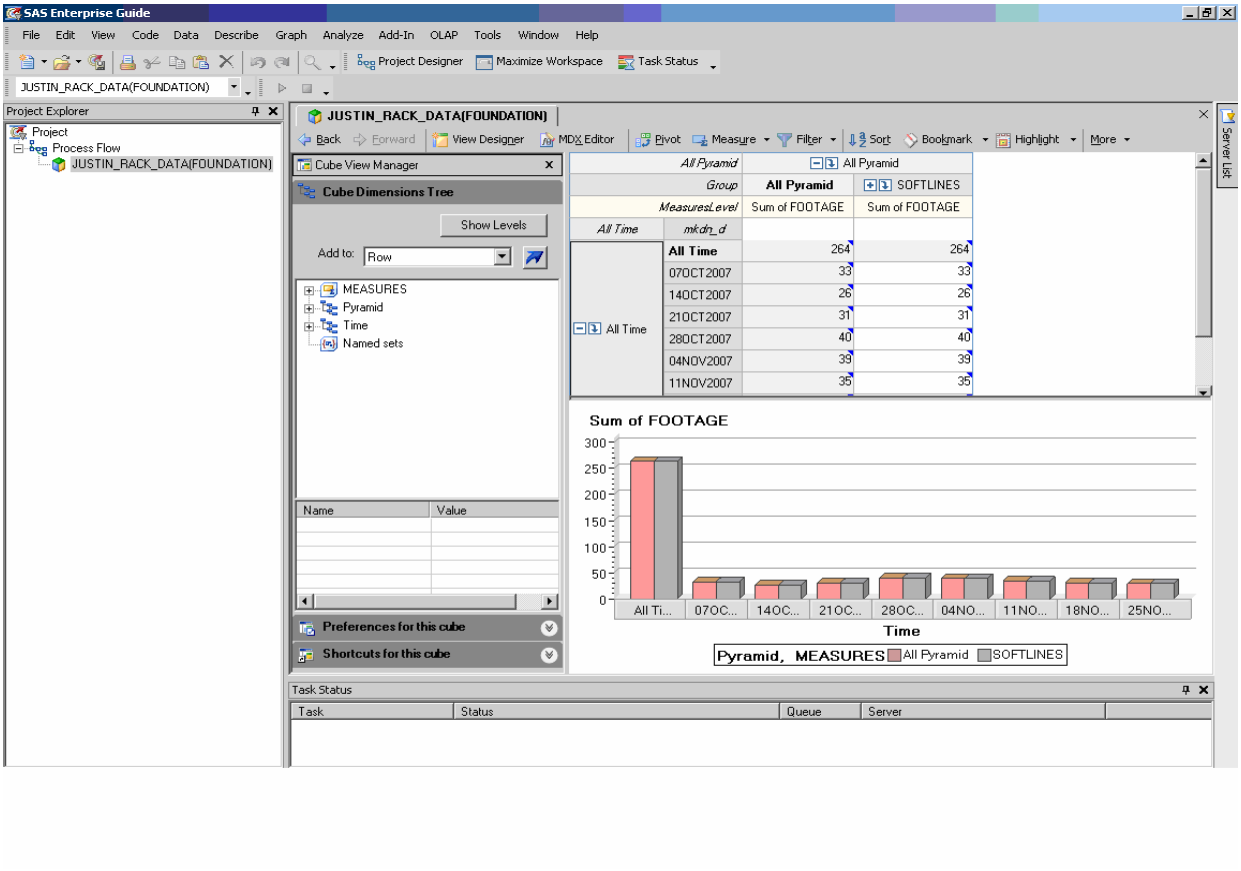


Figure 7: Viewing SAS OLAP Cube in SAS EG

The SAS OLAP Cubes can be viewed like any SAS SPD data set; drill-thru and pivot options appear to the user as new buttons in their SAS Enterprise Guide Session. Where otherwise multiple querying off several tables would have been used, only the click of the mouse is needed for the user to access different levels, or pyramids, of data. Time is immediately saved, and any analysis on the data can be done with more confidence and patience.

BUSINESS INTELLIGENCE TEAM

For nearly all of the business decisions that are made by our larger group of SAS users, there has to exist some sort of statistical information that will give their sound decisions credibility. This is where a Business Intelligence Team comes in. A BI Team can act as the gatherers of any particular data that's needed, whether it be ad-hoc or routine-based; they can serve as a type of Admin to SAS SPD Server data sets, and the SAS software used to compile it; and, although not always common place, they can certainly serve as trainers to the larger group on SAS software. As such mentioned, duties of a BI Team can be limitless, though for us it has been mostly trended towards data gathering. A BI Team will know the data warehouse more detailed over the other users in the group, just from accessing and maintaining it so commonly, that it is not unexpected for them to be called the Advanced Data Gatherers of the system. This can also include exploring other avenues of data accessing other than the traditional warehouse—and definitely does for us. SAS EG is a great tool in the fact that it allows user to thread together information from a wide range of sources.

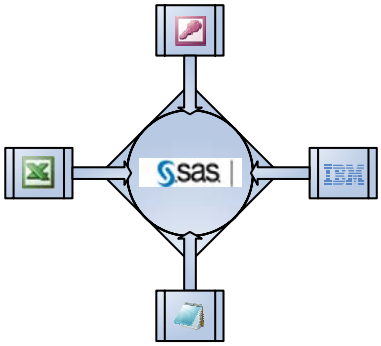


Figure 8: SAS Read-In to multiple forms of data

A user can read in files formats such as .mdb (Access), .xls (Excel), .csv, and do more uniquely challenging data gathering techniques such as submitting SAS EG jobs that access data housed in IBM mainframes. Not surprisingly, it has become the norm to see many SAS EG Projects of BI Team Members that will read an Excel file, combine data from an outside mainframe, and link everyday SAS SPD Server tables all together in the steps from one single process flow.

TRAINING

Getting free time outside of normal ad-hoc/routine-based duties seems to be the single biggest eliminating factor in terms of carving time to a BI Team's more advanced data gathering methods. Training the users supported by the BI Team with independent knowledge of the system has become a necessity.

Figure 9: Diploma for internal SAS training

Implementing a "SAS University", as we have done, can make it mandatory for regular users to know enough of the SAS Tools and any accompanying warehouse(s) to work freely from their BI Team Members; this parallel work atmosphere will open the doors to more complex projects on the BI Team, and should ultimately advance the whole group through the long run. Simple tests has been our method of choice for declaring graduation to this "SAS University", but that is not to say that continued learning hasn't been routinely presented to users when refreshers, or informative new best practices, spring up; half-sheets with new SAS EG Project steps or new tips-and-tricks is a good way to keep regular users up-to-date with all things important. While this significance of keeping regular SAS users in-the-know about the global benefits of SAS is beneficially important, it is extremely imperative that a BI Team's support to its user's exploration of SAS EG is never momentarily seized; a BI Team has to be on-call to the demands of their users 24-7 while the system is being used—that's just how it has to go when BI serves as the Advance Data Gathers, otherwise problems are quickly sure to rise.

DATA WAREHOUSE ADMINISTRATOR

While data gathering is sure to be the most crucial aspect of a BI Team—this has been made clear—something that it shouldn't do is completely belittle the Administrative piece that needs part of the pie. Administration is a must, as well. To monitoring warehouse space, ETL Processes, and CPU usage (monthly/current), to adding new users to the UNIX environment, BI Team Members can serve as SAS Admins on much smaller, more personal scale. Just having the access to web-based system server monitors allows the up-keeping of the SAS Environment to be quite simple.

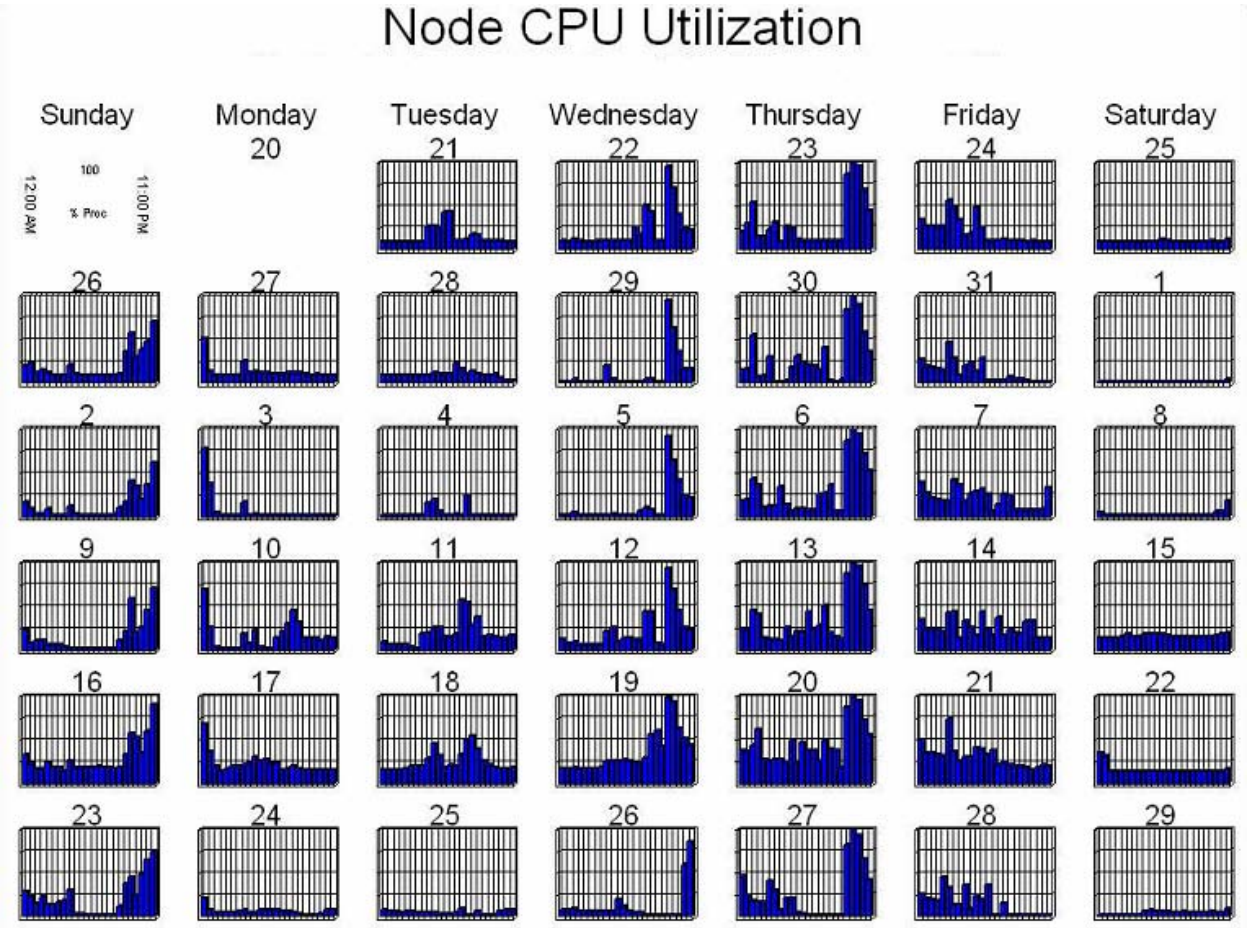
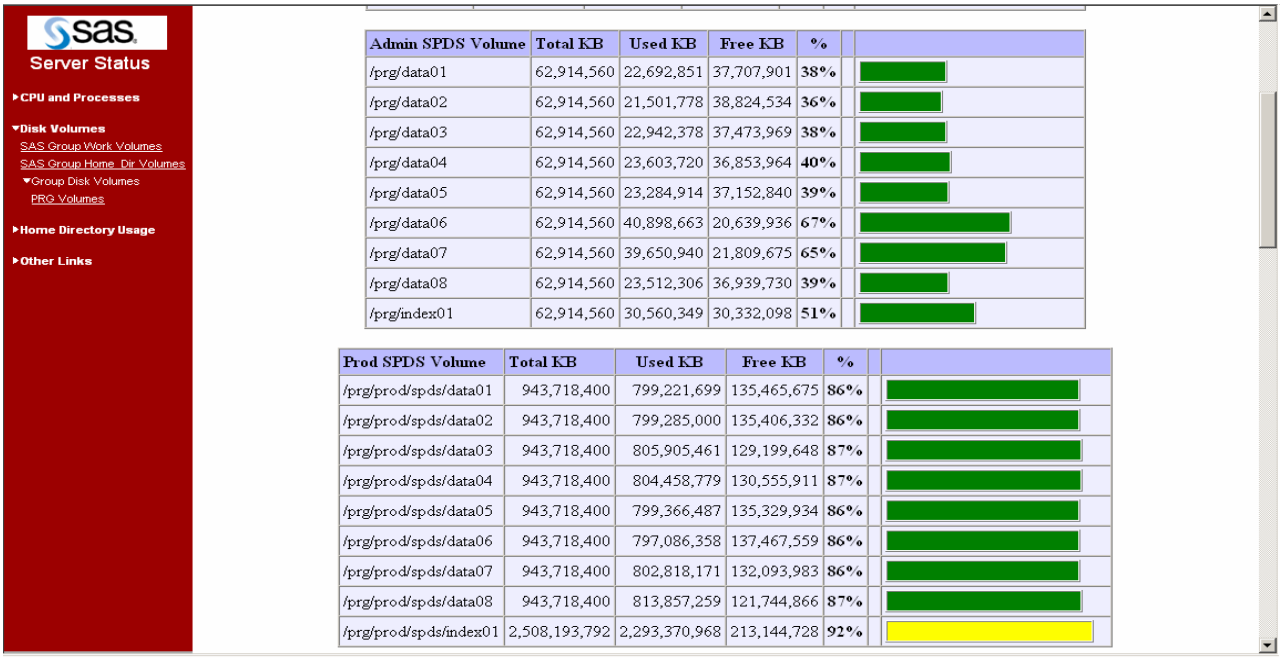
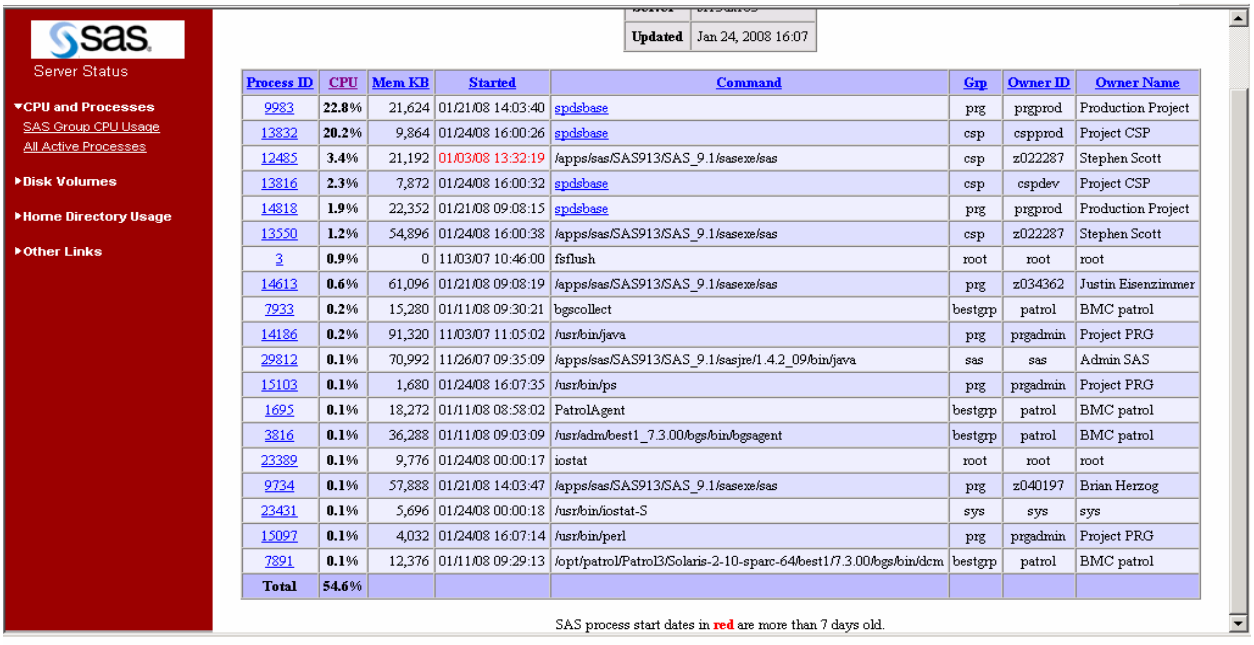


Figure 10: Our division's CPU Use from monthly Capacity Planning meeting with SAS Admins.



Admin SPDS Volume	Total KB	Used KB	Free KB	%
/prg/data01	62,914,560	22,692,851	37,707,901	38%
/prg/data02	62,914,560	21,501,778	38,824,534	36%
/prg/data03	62,914,560	22,942,378	37,473,969	38%
/prg/data04	62,914,560	23,603,720	36,853,964	40%
/prg/data05	62,914,560	23,284,914	37,152,840	39%
/prg/data06	62,914,560	40,898,663	20,639,936	67%
/prg/data07	62,914,560	39,650,940	21,809,675	65%
/prg/data08	62,914,560	23,512,306	36,939,730	39%
/prg/index01	62,914,560	30,560,349	30,332,098	51%

Prod SPDS Volume	Total KB	Used KB	Free KB	%
/prg/prod/spds/data01	943,718,400	799,221,699	135,465,675	86%
/prg/prod/spds/data02	943,718,400	799,285,000	135,406,332	86%
/prg/prod/spds/data03	943,718,400	805,905,461	129,199,648	87%
/prg/prod/spds/data04	943,718,400	804,458,779	130,555,911	87%
/prg/prod/spds/data05	943,718,400	799,366,487	135,329,934	86%
/prg/prod/spds/data06	943,718,400	797,086,358	137,467,559	86%
/prg/prod/spds/data07	943,718,400	802,818,171	132,093,983	86%
/prg/prod/spds/data08	943,718,400	813,857,259	121,744,866	87%
/prg/prod/spds/index01	2,508,193,792	2,293,370,968	213,144,728	92%



Updated	Jan 24, 2008 16:07

Process ID	CPU	Mem KB	Started	Command	Grp	Owner ID	Owner Name
9983	22.8%	21,624	01/21/08 14:03:40	spdsbase	prg	prgprod	Production Project
13832	20.2%	9,864	01/24/08 16:00:26	spdsbase	csp	cspprod	Project CSP
12485	3.4%	21,192	01/03/08 13:32:19	/apps/sas/SAS913/SAS_9.1/sasexe/sas	csp	z022287	Stephen Scott
13816	2.3%	7,872	01/24/08 16:00:32	spdsbase	csp	cspdev	Project CSP
14818	1.9%	22,352	01/21/08 09:08:15	spdsbase	prg	prgprod	Production Project
13550	1.2%	54,896	01/24/08 16:00:38	/apps/sas/SAS913/SAS_9.1/sasexe/sas	csp	z022287	Stephen Scott
3	0.9%	0	11/03/07 10:46:00	fsflush	root	root	root
14613	0.6%	61,096	01/21/08 09:08:19	/apps/sas/SAS913/SAS_9.1/sasexe/sas	prg	z034362	Justin Eisenzimmer
7933	0.2%	15,280	01/11/08 09:30:21	bgscollect	bestgrp	patrol	BMC patrol
14186	0.2%	91,320	11/03/07 11:05:02	/usr/bin/java	prg	prgadmin	Project PRG
29812	0.1%	70,992	11/26/07 09:35:09	/apps/sas/SAS913/SAS_9.1/sasjre/1.4.2_09/bin/java	sas	sas	Admin SAS
15103	0.1%	1,680	01/24/08 16:07:35	/usr/bin/ps	prg	prgadmin	Project PRG
1695	0.1%	18,272	01/11/08 08:58:02	PatrolAgent	bestgrp	patrol	BMC patrol
3816	0.1%	36,288	01/11/08 09:03:09	/usr/adm/best1_7.3.00/bgs/bin/bgsagent	bestgrp	patrol	BMC patrol
23389	0.1%	9,776	01/24/08 00:00:17	iostat	root	root	root
9734	0.1%	57,888	01/21/08 14:03:47	/apps/sas/SAS913/SAS_9.1/sasexe/sas	prg	z040197	Brian Herzog
23431	0.1%	5,696	01/24/08 00:00:18	/usr/bin/iostat-S	sys	sys	sys
15097	0.1%	4,032	01/24/08 16:07:14	/usr/bin/perl	prg	prgadmin	Project PRG
7891	0.1%	12,376	01/11/08 09:29:13	/opt/patrol/Patrol3/Solaris-2-10-sparc-64/best1/7.3.00/bgs/bin/dcm	bestgrp	patrol	BMC patrol
Total	54.6%						

SAS process start dates in red are more than 7 days old.

Figure 11/12: SAS Server Monitor for our group

A space-used homepage can provide worry-free insight into current storage capacity, and its constant monitoring can benefit predictions of future growth estimations; this same web-based system can even play Big Brother to active or inactive SAS processes out there, shedding unseen light to present clean-up possibilities, or your co-worker's CPU hogging.

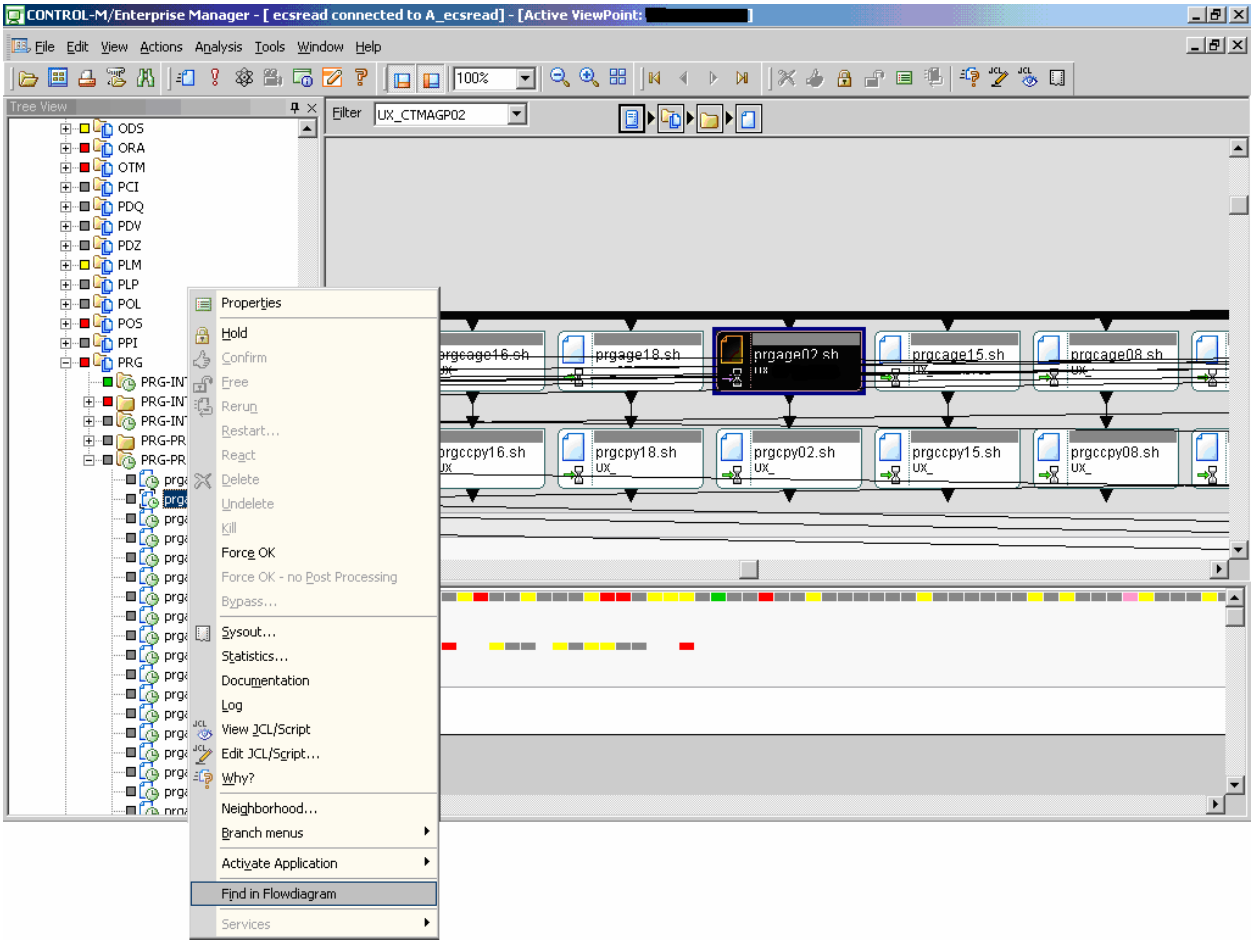


Figure 13: BMC Control-M Enterprise Job Scheduler showing Unix SAS jobs for production

Such admin benefits can, likewise, come from Control-M. The weekly Unix process that appends critical warehouse data to our warehouse can be check-off as completed, or dipped into for bugs and abends—all being done with web-based point-and-click system monitors; this kind of guarantee to your data's retrieval in the small window of time that it needs pulling can provide comfort for other efforts.

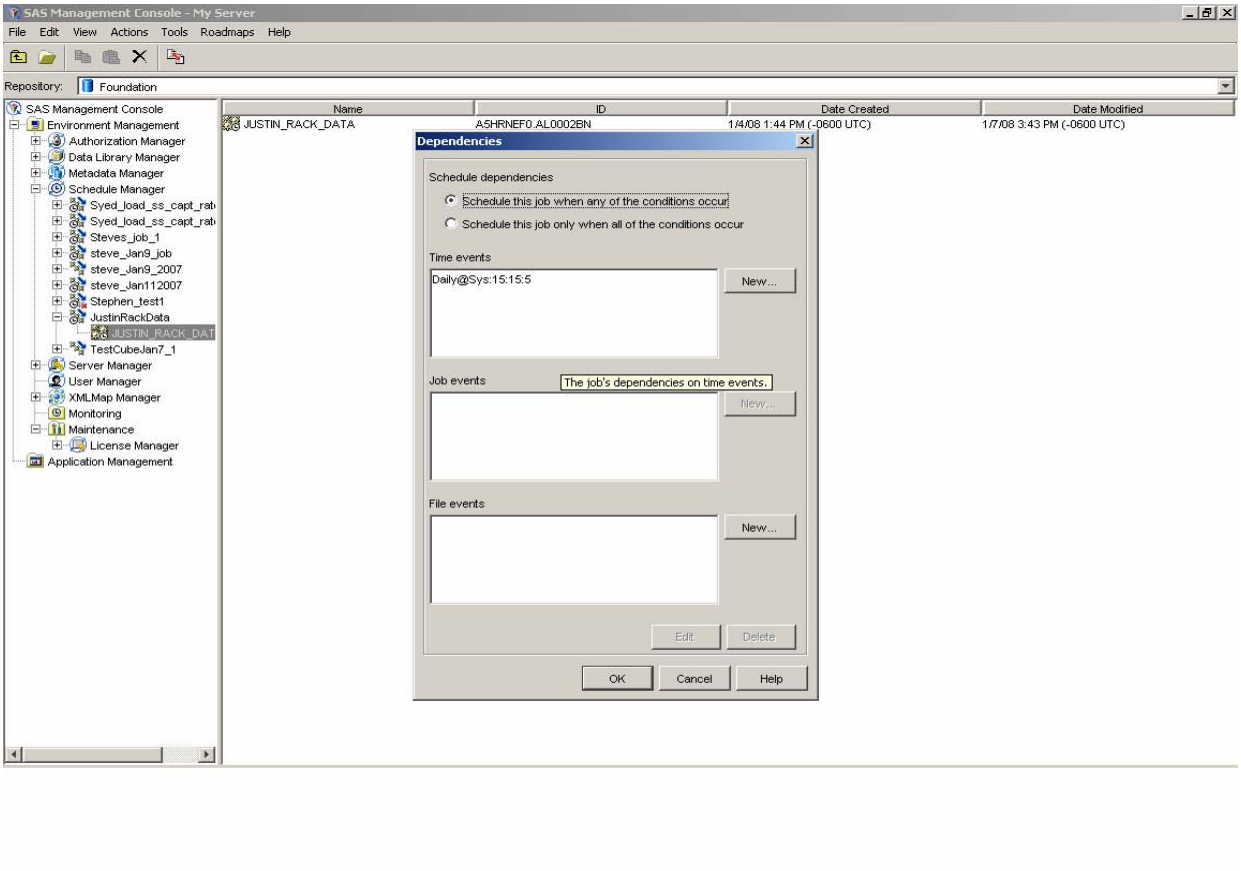


Figure 14: SAS Management Console currently needed for scheduling SAS DI studio jobs

As such another effort could be the power user scheduled of SAS Processes that have been constructed solely out of SAS Data Integration Studio, which, like the weekly mainframe extracts, can augment and append data as needed; this self-storing of information can give benefits through flexible code logics (SAS DIS) and window of run times (SAS Scheduler), dissimilar to automated processes and their hands-off superiorities. Though data can warehouse in either of these two scenarios (production controlled or power user scheduled), its credibility only forms as users begin to apply it to questions; here's where user accessibility plays its part.

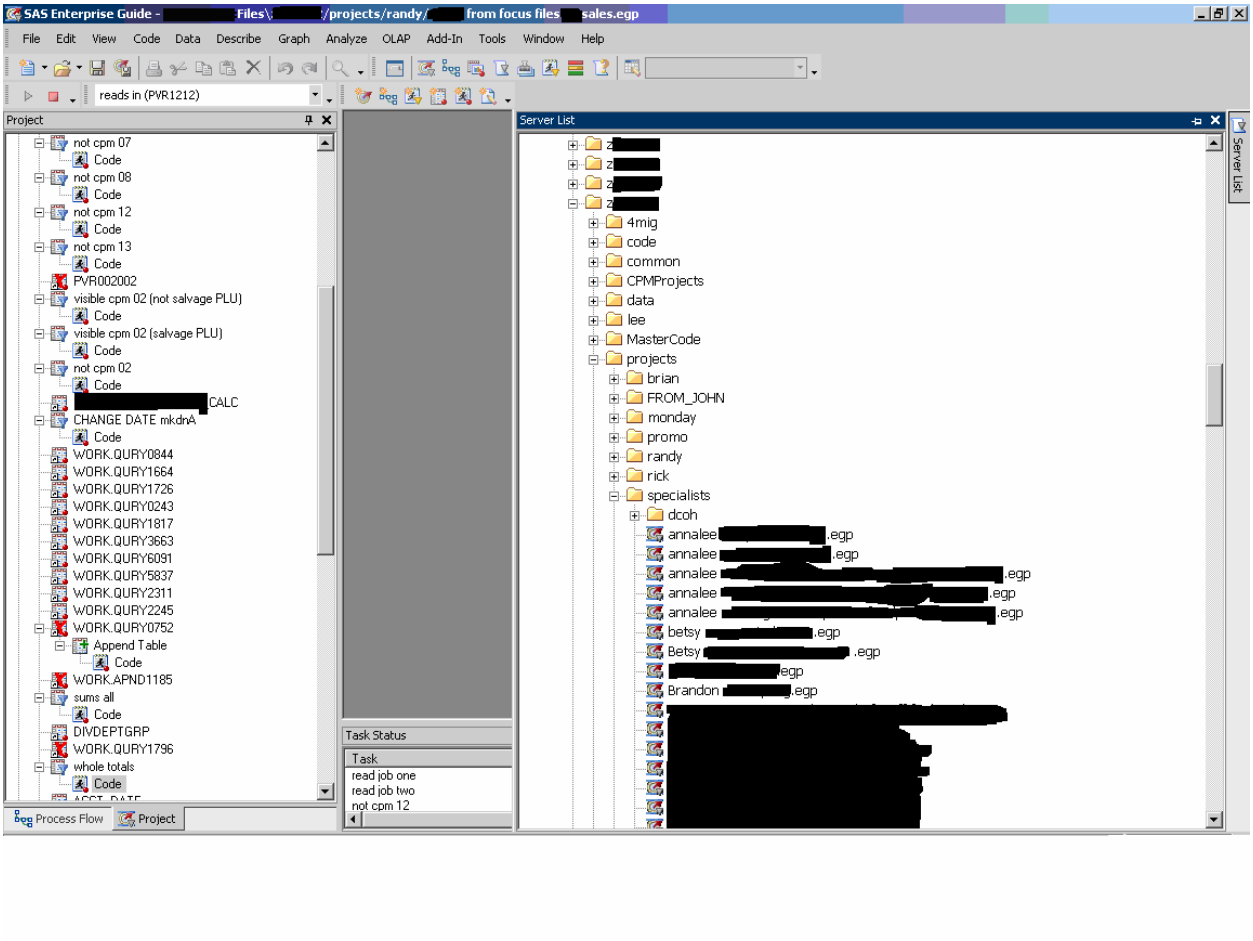


Figure 15: SAS EG Unix directory folders for business user's ID

The BI Team's Admin duties happen to widely cover the basics of introductions for new users. Granting SAS Unix access, setting up SAS User Directories, and allowing mainframe privileges are the standard blueprints to giving users their data fixes for their own hands—when luck plays its part, as it has for us, most these can be simple Admin Service Requests, leaving only the mattering of filling out a form the ultimate task before data and business questions can be one.

I.T. Perspectives: Designing and Building the SAS Scalable Performance Data Server areas:

Challenge:

The challenge within corporate I.T. departments with the production level usage of SAS software is the integration of the systems themselves into the overall approved development architecture of the corporation. With that accomplished the next challenge is the development of a standard modular system that is restart able and supportable that runs within the approved corporate production IT architecture and meets all Levels of Service agreements. All of this is occurring within an Enterprise system that is changing and evolving into new standards and architectures all the while based on old and new legacy systems and approved processes.

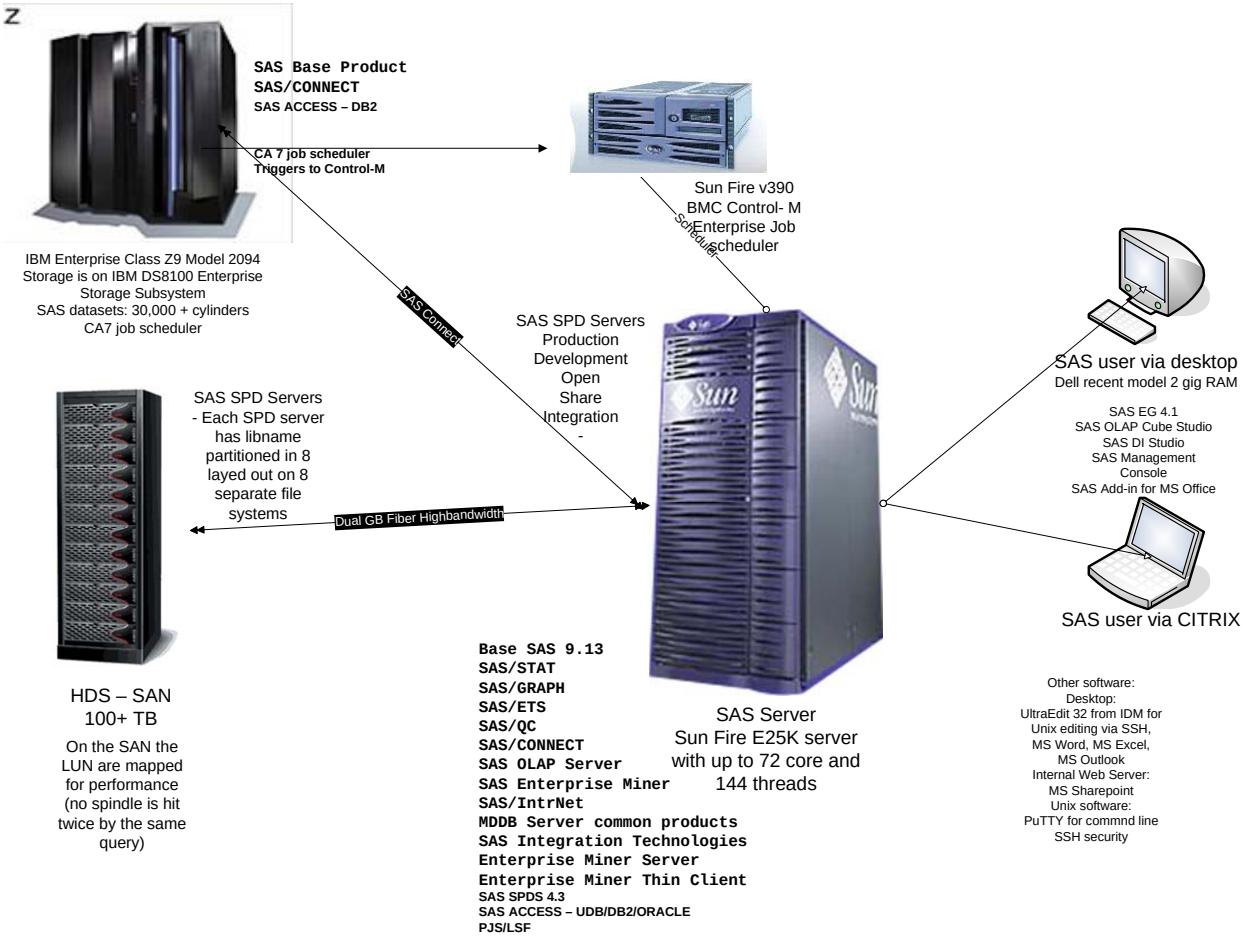


Figure 16: our SAS Architecture with SAS Scalable Performance Data Server

TECHNICAL DESIGN: HARDWARE SPECIFICATIONS
1. Sun Fire E25K (high end server for SAS)

This is always in evolution and we upgrade when appropriate. We moved from Solaris 8 to Solaris 10 and from Sun Spark III to Sun UltraSparc IV+ in Oct. 2006. At the same time we moved from a shared area where our group competed for 48 CPU with the other groups to a separate domain where we have the SAS licensing equivalent of 24 CPU which is really 12 Sun Spark IV+ with 24 threads that we share with another partner group. This system is extremely powerful so that for us the 12 Sun Spark IV easily outperformed the 48 Spark III and our run time was reduced significantly.

This is the overall configuration summary for the Sun Fire E25K Server that scales up to 72 UltraSPARC IV+ with 144 threads. We continue to grow and will reach capacity by the time a new model is available.

domain	Model #	Num. CPU	CPU Speed	Memory GB	OS Version	SAN storage: Terabytes
a	SunFire E25K	16	1.3 GHz	64	5.1	4
b .	SunFire E25K	40	1.5 GHz	192	5.1	50
c	SunFire E25K	32	1.5 GHz	192	5.1	15
d	SunFire E25K	24	1.5 GHz	96	5.1	31
		Total CPUs		Total GB		Total TB
		112		544		100

Table 1: Sun Server Architecture for SAS usage with SAS Scalable Performance Data Server

All SAS storage is on a dedicated SAN that is provisioned from a Hitachi USP V. It is RAID 5 7D+1P.
Above are the approximate storage totals by server. Previously we had EMC storage.

2. IBM Enterprise Class Z9 Model 2094 (high end mainframe with SAS access to our DB2 and other mainframe file system source data)

Our source data is accessed on the most powerful class of IBM mainframes the Enterprise Class Z9 Model 2094 with 5 processors and 3 parallel sysplex with multiple lpar. SAS is licensed to run on 2 processors and is enabled on 3 logical partitions.

This platform is also very powerful and effective. We have the best hardware available and our performance is excellent. We use 30,000 cylinders to hold our weekly data stored and accessed on
IBM DS8100 Enterprise Storage Subsystem (type: 2107). Each of the processors (F1-->F5) have multiple physical channel paths to this DASD machine through a switched fiber network topology of 16 2GB/4GB FICON Director

ARCHITECTURE OF THE SAS SCALABLE PERFORMANCE DATA SERVER IN LINE WITH CORPORATE STANDARDS AND METHODOLOGY

The following separate SPD Server domains were created with their own space allocations. They each are separate and can be stopped or started independently if a bounce is needed to remove users. We currently support in our division 2 groups PRG and CSP. Each of these will have a similar independent set of SPD Server libraries.

Enterprise owns, read only for business and IT: Integration, Stage, Production, History
Business owns, read write for business: Open, Share
IT owns, read write for IT Development

At the end of the project the Development area was given to the Business. The Stage area was released and reallocated to the Production area.

STORAGE SPACE SIZE OF THE SAS SPD SERVER DOMAINS (APPROXIMATE)

PRG:
Production: 20 terabyte
Integration: 1 terabyte
Open: 1 terabyte (business read write area)
Share: 1 terabyte (business read write area)
Development: 1 terabyte (IT read write area, transition to business read write)
Work: 2 terabyte (shared work areas used by all SAS EG user community)
Home: 500 gig (each user has space here for projects and data)

Please note that compression is on and our data is compressed at a ratio of close to 50%. This means that the real data size of our system is double what we see here.

DEVELOPMENT METHODOLOGY IN LINE WITH CORPORATE STANDARDS AND BUSINESS PRACTICES

1: SAS ENTERPRISE GUIDE

2. SAS DATA INTEGRATION STUDIO AND SAS OLAP STUDIO

3. EDITING TOOL FOR SCRIPTS: ULTRAEDIT-32

When we began this project we received approval to develop and test the system at the unit test level with SAS Enterprise Guide. That is the same process used by the business to access the data. This facilitates communication and is used corporate wide since we have an unlimited Enterprise BI license for SAS EG. It is easy for the SAS Admin to support and is also a shared point of communication between the developers in the Information Technology department and the Business departments.

Only later in the evolution of these systems did we receive permission to bring in SAS Data Integration Studio with SAS OLAP Cube Studio as a business level tool for the integration of data. The I.T. standard for the ETL processes for other data warehouses is another product. However the approved tool for the business is SAS EG for ad hoc jobs and SAS Data Integration Studio for large jobs to be scheduled by the business user via Platform LSF Job Scheduler. When the I.T. development team met at the start of this project with the corporate Enterprise Architecture teams it was agreed that the best development approach in our case was to use the SAS Enterprise Guide for development with the final code to be stored in Unix or mainframe directories and submitted by the approved job schedulers CA7 for mainframe and BMC Control-M for Unix and other servers. At this time if we want to use SAS Data Integration Studio to create code we can also do so. We can also write our own code in the code window provided in the SAS tools or in UlraEdit-32 editor. The code is tested via SAS EG or if large it can be run in UNIX via scripts.

Our Technical Development/Integration/Production Structure

In the screen print on the Server list are examples of the file systems where we keep our final code and our code at the level of development and test.
There are 3 main UNIX directories for the final technical development and production functions: Development, Integration, and Production.

As we open each one you see they contain similar subdirectories:

1. CMSCRIPT: the script for Control-M.
2. CNTL: SAS library names, SAS options and common shared scripts.
3. CODE: the SAS code from whatever method it is coded or generated,
4. LOG: all logs are time stamped and stored.
5. LST: some control reports are placed here,
6. SCRIPT: a UNIX script for every job is here.

The main directory for the initial development is the individual User ID which in this example is Z123456 which is the directory where the Log shown is from and where the Project shown is from.
Here in this screen shot is a composite of how we develop the code and test it and analyze the resulting data.
Notice that the log shows the code which has an ‘include’ for the libname that is commented out to run in SAS EG. It also has macro variables so we can read and write to different libraries depending on our level of development.

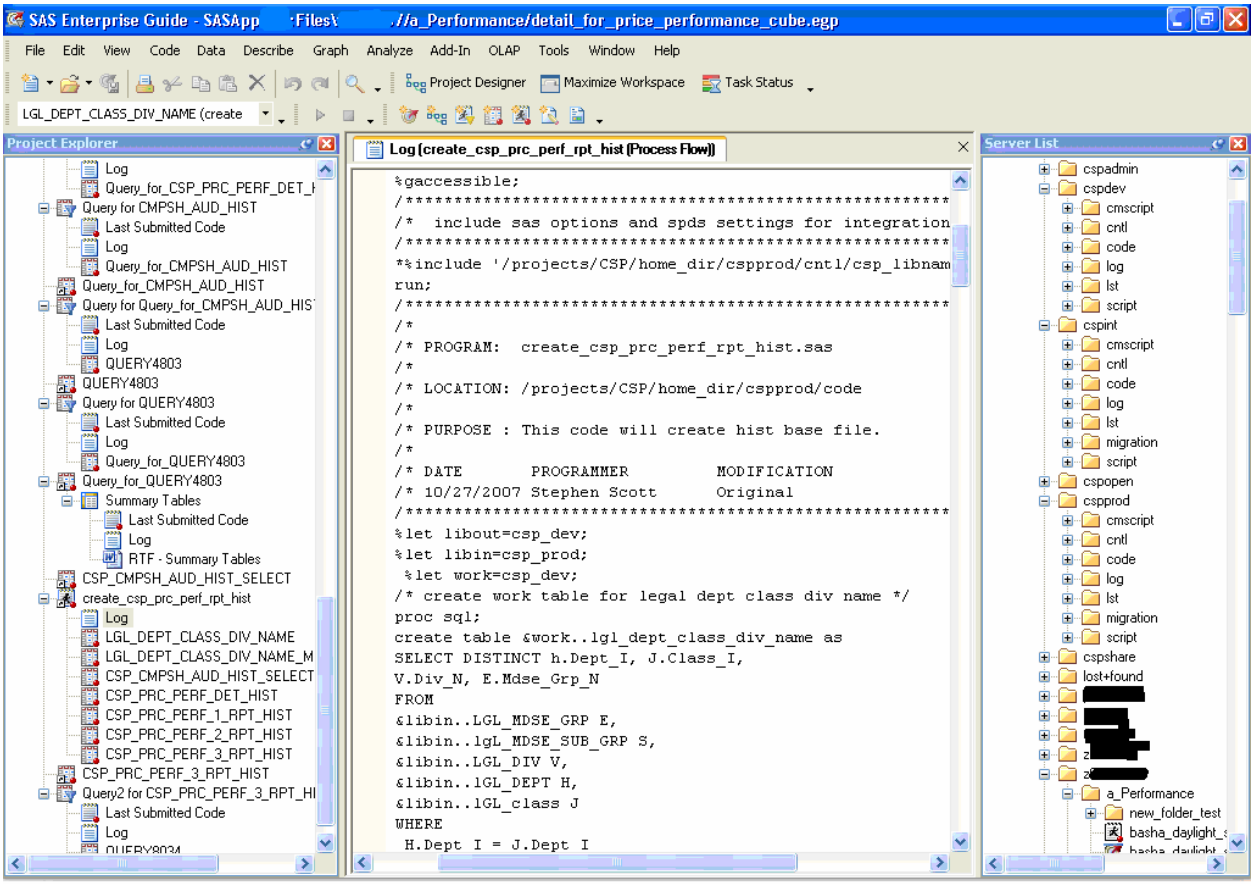


Figure 17: SAS Technical Development Methodologies and Architecture

Once the code is developed and tested the log shown above is stored and used as proof to change management for the move to Integration. The example below shows the logs from Integration. These are needed as proof for change management in order to move the code to Production. They are stored in HP Service Desk which is the approved Change Management tool.

Here you can see that our logs are produced by an automated script that creates the subdirectory for the year and the month and stamps each job in the log with the time it ran. This is kept for the life of the data.

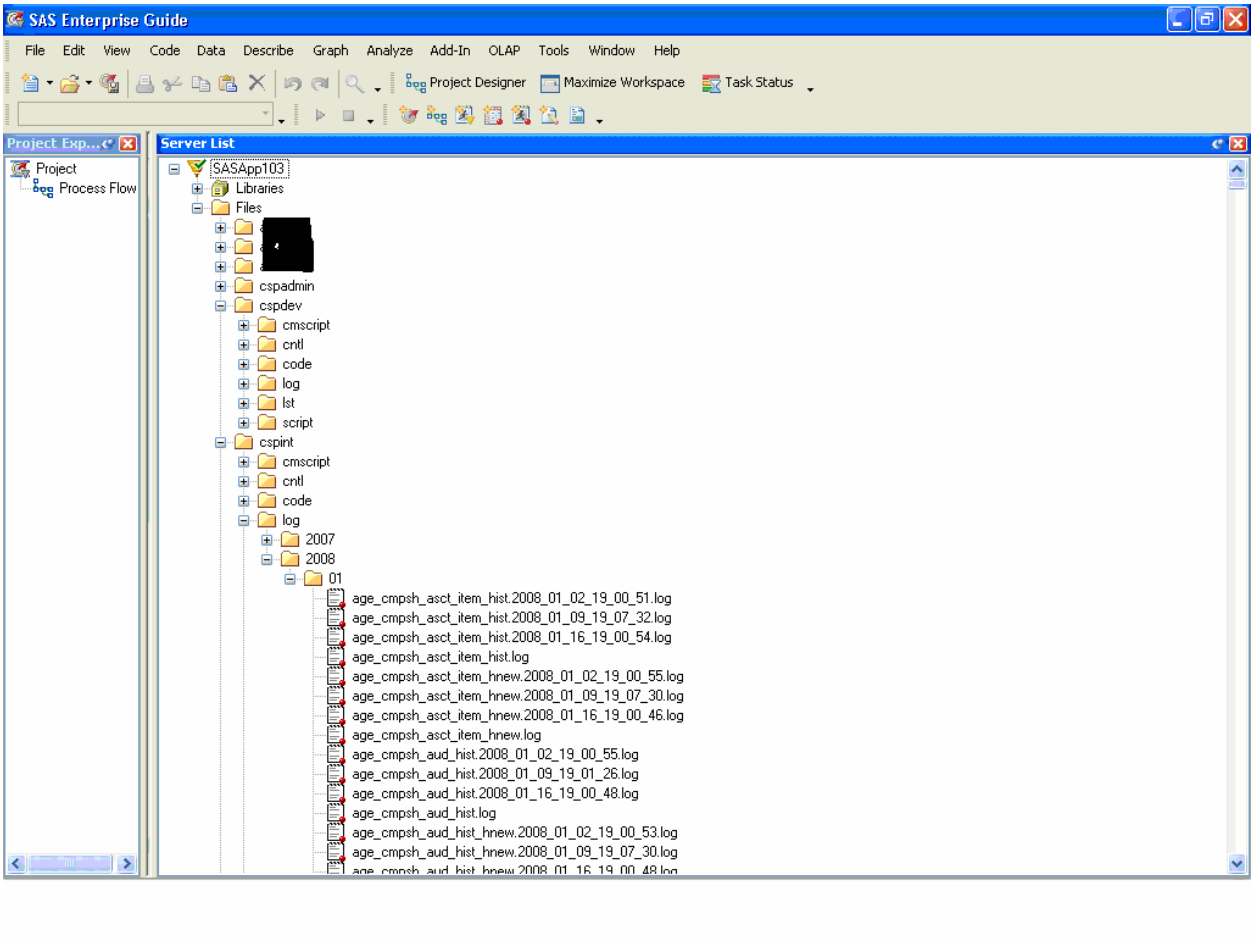


Figure 18: SAS Log Directory Storage for 2 to 5 year life of data

IT Partnership Teams:

Other partner teams with the Development Team included: SAS Admin, Info. Security, Unix Admin., Capacity Planning, Production Control, ESS Support, Mainframe support, Change Management, SharePoint Admin, and Software Deployment.

At different times each of these teams played a key role in this project and they continue to do so.

HOW WE DEFINE METADATA FOR THE SAS SPD SERVER LIBRARIES.

The SAS Management Console is used to manage the SPD libraries and user security.

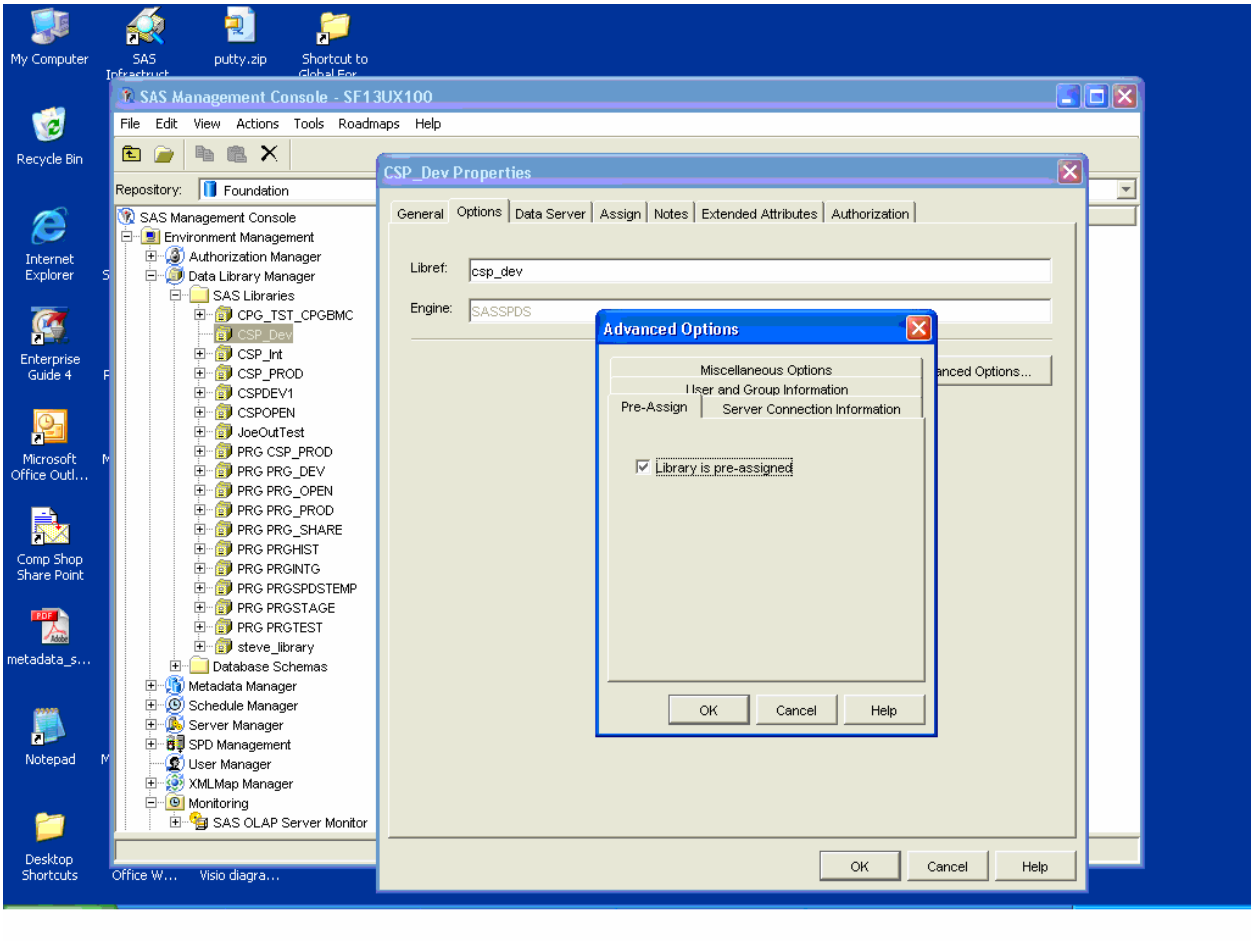


Figure 19: SAS Management Console with SPD Server library pre-assigned

PERFORMANCE OF SAS SPD SERVER AND THE COMING CONVERSION TO CLUSTER TABLES:

Here are some examples of our use of the best practices for adding rows to large base tables without cluster architecture which is the append procedure. I will first describe one of our large tables that in its various forms contain 700 million to 2.8 billion rows. It is also very wide with 92 columns and a width of 967 bytes. When new data comes in we need to add it to our historical table. In Figure 20 the new data is 23 million rows. We index the new data to prepare it. Then we append it to the historical data which is close to 700 million rows. There is a unique index on the data so the appending is intensive. This is the same table accessed in Figure 3.

An advantage of the cluster is that the append steps would be eliminated saving 54 real minutes on the 'accum' table and 21 real minutes on the 'calc' table. If you multiply this by another 10 large tables each with an 'accum' and a 'calc' variation this approaches savings of 10 hours of real time which is significant and the total CPU time savings is 180 minutes which is helpful and allows more processing from the business users to occur in the time frame.

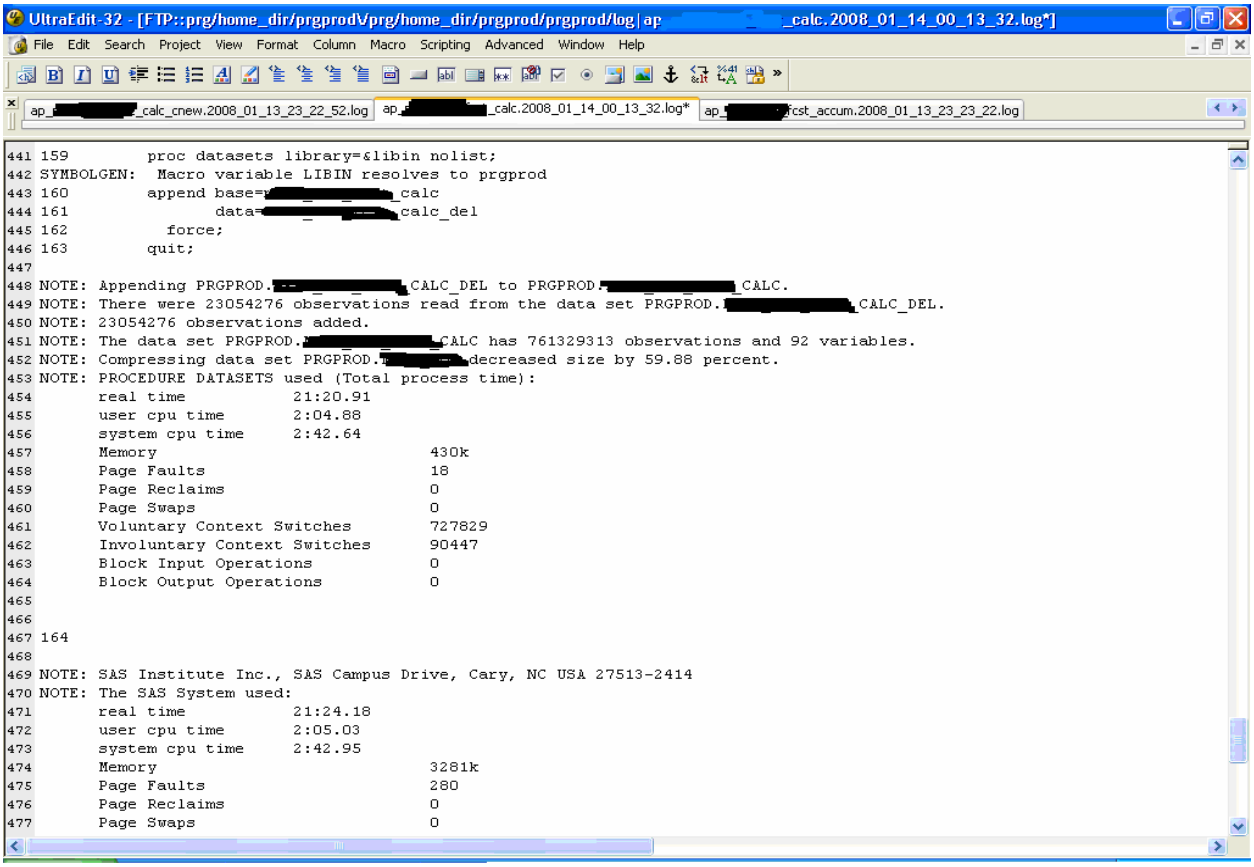


Figure 20: Appending Weekly Delta to Five Year Table in SAS SPD Server

Append rows: 23,054,276
Append size: 21 GB
Base Rows: 761,329,313
Row bytes: 967
Total size in bytes: 736205445671
GB: 685
TB: .669
Total CPU to append data: 4:47.98
Real time to append data: 21:20
Note: The append process is also maintaining the indexes. Comparing this to a different version of the table with 2.9 billion rows and appending 56 million rows there is a Ratio of about 2.5 X Real time and 2.5 X CPU time and 2.4 X rows appended. The size is close to 56 GB appended to 2.5 TB with 15 indexes being maintained.

Here is a description of the SAS SPD Server table from the proc datasets procedure in SAS EG:

Data Set Name	PRG_PROD.XXXX_CALC	Observations	761329313
Member Type	DATA	Variables	92
Engine	SASSPDS	Indexes	15
Created	Wed, Jan 09, XXXX 07:02:29 PM	Observation Length	967
Last Modified	Monday, January 14, XXXX12:13:47 AM	Deleted Observations	0
Protection		Compressed	YES
Data Set Type		Reuse Space	NO
Label		Point to Observations	YES
Data Representation	Default	Sorted	NO
Encoding	latin1 Western (ISO)		

Engine/Host Dependent Information	
Blocking Factor (obs/block)	33
ACL Entry	YES
ACL Default Access(R,W,A,C)	(N,N,N,N)
ACL Group Access(R,W,A,C)	(N,N,N,N)
ACL User Access(R,W,A,C)	(Y,N,N,N)
ACL UserName	USER
ACL OwnerName	PROD
ACL GroupName	USERGRP
Data set is Ranged	NO
Data set is a Cluster	NO

Index	Unique Option	# of Unique Values	Variables
OUT_D		330	
UNIQ	YES	761329313	Var1 Var2 Var3 Var4 Var5 Var6
Var1		236	
Var2		131	
Var3		1789	
Var4		437	
Var5		13	
Var6		54	
Var7		9	
Var8		65486	
Var9		175	
Var10		1116	
Var11		13	

We run the index so that they are created in parallel. This is very fast and takes all of our CPU which is what we want since it is running on the scheduler during off hours for both onshore and offshore resources.

```
proc datasets lib=&libname nolist;
modify &dsname;       /* XXXX_CALC;    */
index create Var1;
index create Var2
index create Var3
index create Var4;
index create Var5;
index create Var6;
index create Var7;
index create Var8;
index create Var9;
index create Var10;
index create Var11;
index create UNIQ=( Var1 Var2 Var3 Var4 Var5 Var6;) /unique ;
quit;
```

Future Direction:

Here is a list of some large tables that we plan to turn into cluster tables. The weekly appending to these tables and the weekly recycle of history can be eliminated by the cluster architecture for tables that find this necessary for the business architecture. We have written a detailed proposal with the help and support of the business. In my testing of a 100 million row table with cluster and without there was also a 10% improvement in typical query time for the business user with the cluster table.

Table	Rows	Lng	Size	Cmp	Space Saved	Real Space
Table 1	3,035,949,696	947	2,875,044,362,112	0.607	1,745,151,927,802	1,129,892,434,310
Table 2	4,640,937,516	453	2,102,344,694,748	0.607	1,276,123,229,712	826,221,465,036
Table 3	4,423,225,228	388	1,716,211,388,464	0.6252	1,072,975,360,068	643,236,028,396
Table 4	768,732,783	967	743,364,601,161	0.5988	445,126,723,175	298,237,877,986
Table 5	765,526,005	967	740,263,646,835	0.5988	443,269,871,725	296,993,775,110
Table 6	761,329,313	967	736,205,445,671	0.5988	440,839,820,868	295,365,624,803
Table 7	694,321,175	967	671,408,576,225	0.5981	401,569,469,440	269,839,106,785
Table 8	1,073,705,803	392	420,892,674,776	0.6224	261,963,600,781	158,929,073,995
Table 9	1,062,865,895	392	416,643,430,840	0.6224	259,318,871,355	157,324,559,485
Table 10	1,028,564,770	392	403,197,389,840	0.6224	250,950,055,436	152,247,334,404
Table 11	950,141,757	275	261,288,983,175	0.4557	119,069,389,633	142,219,593,542
Table 12	1,179,634,692	144	169,867,395,648	0.3617	61,441,037,006	108,426,358,642
Table 13	1,215,105,579	156	189,556,470,324	0.4557	86,380,883,527	103,175,586,797
Table 14	1,182,932,259	118	139,586,006,562	0.345	48,157,172,264	91,428,834,298
Table 15	1,170,283,502	118	138,093,453,236	0.345	47,642,241,366	90,451,211,870
Table 16	1,087,795,673	118	128,359,889,414	0.3444	44,207,145,914	84,152,743,500
Table 17	450,705,118	275	123,943,907,450	0.4557	56,481,238,625	67,462,668,825
Table 18	116,860,224	621	72,570,199,104	0.3463	25,131,059,950	47,439,139,154
Table 19	113,081,439	621	70,223,573,619	0.3463	24,318,423,544	45,905,150,075
Table 20	113,045,546	621	70,201,284,066	0.3463	24,310,704,672	45,890,579,394
Table 21	111,348,060	621	69,147,145,260	0.3463	23,945,656,404	45,201,488,856
Table 22	111,154,509	621	69,026,950,089	0.3463	23,904,032,816	45,122,917,273
Table 23	110,998,419	621	68,930,018,199	0.3463	23,870,465,302	45,059,552,897

Table 2: SAS SPD Server tables with compressed size

CONCLUSION

To create a highly effective data integration system and data warehouse, the IT and business relationship is important. The business client defined the project clearly and produced comprehensive and specific requirements. IT implemented several technologies including key SAS products to develop an efficient and reliable warehouse. The result is a SAS Scalable Performance Data Server based warehouse that is used daily to answer business questions, produce forecasting models and what-if scenarios, etc. All of this allows the organization to respond to the business environment quickly, creating efficient and effective solutions.

CONTACT INFORMATION

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

Stephen C Scott
scott_stephen@yahoo.com
Justin Eisenzimmer
eisenzjw@comcast.net
Brian Herzog