

SAS/CONNECT® - Trials and Triumphs of Migrating from Batch to Interactive Processing in an IBM® Mainframe Environment

Joseph M. Braun, Purdue University, West Lafayette, IN
 Michael S. Hines, Purdue University, West Lafayette, IN
 Shawn L. Williams, Purdue University, West Lafayette, IN

ABSTRACT

Easy, direct access to data by users is a key business requirement. With windows-based Graphical User Interface (GUI) tools being the norm, batch-processing tools do not meet today's expectations of the non-programmer user. However, there is no reason that "power users" shouldn't have easy to use tools as well.

SAS System for personal computers (PC/SAS) coupled with SAS/CONNECT is a powerful pair of tools for gaining interactive access to mainframe data. This presentation will identify the challenges and rewards involved in introducing interactive processing into a primarily batch MVS reporting environment. The challenges included securing approval for licensing, and installation, configuration and testing of SAS/CONNECT. Additionally, the SAS/CONNECT team re-created, in the TSO environment, many of the standard utilities that existed in the batch world. The project team also prepared training resources to educate users in the basic techniques of the new tools.

The predominate triumph is that users and programmer/analysts alike now share GUI access to data that had previously been "locked" up on the mainframe. Users also acquired a tool for easily mixing and moving data across operating platforms and choosing where processing occurs. Finally, this project was a catalyst that brought together SAS users from different departments to rejuvenate our in-house user group.

INTRODUCTION

Even today, most of Purdue University's Administrative Management Information (MI) mission critical processes (Student Processes, Purchasing, Payroll etc.) are accomplished by Cobol programs running on an IBM 390 mainframe. This basic infrastructure has not changed for the past 20+ years. During this time frame, MI staff and Information Analysts, hired by individual departments have hired their own "Information Analysts" to assist with defining needed system modifications, and to generate reports and information from the data on the mainframe computer.

This environment resulted in MI staff who was proficient in IBM JCL®, Cobol, and CICS®. Departmental users were not permitted access to these tools, and were limited to using ROSCOE®, RAMIS®, SPSS®, and several utilities such as SORT, and SELECT. The hardware available was 3270 "dumb" terminals. All processing was done by submission of "batch" jobs, no provision for interactive processing existed.

The data available to departmental analysts was typically a copy of the file used by the online transaction processing system (OLTP). These Remote Job Entry (RJE) files, as

they were called, were updated on a variety of schedules: some nightly, some weekly, some biweekly.

Another important aspect of the data environment was that data was considered to be the property of the departments.

Thus the Office of the Registrar "owned" student records. The payroll office had control of all data needed to correctly pay staff. This mindset forced the MI staff to develop mechanisms, both administrative and technical, for granting and revoking access to data. The result was that a very elaborate set of ROSCOE Program Facility (RPF) were built to provide file creation, editing, and program submission capabilities. Initially these RPF's also controlled access to the data as well.

Administrative Data Processing began providing user access to their data in 1981. The computing environment was ROSCOE terminal access to the MVS system, and remote job entry. At that time RAMIS was the only supported data access tool. The SAS system was introduced into our computing environment in 1983 and was made available as an ad-hoc reporting tool. Some of the SAS users attended SUGI 21, that took place in Chicago, 1996, where they saw the capabilities of the SAS system on a Personal Computer and how to connect to multiple computing platforms using SAS/CONNECT. We were excited about this new tool and pushed the administration to acquire and support the tool.

A great deal of user unrest and dissatisfaction with MI built in the mid-1980's. Because the time frame for developing new or modified reports ranged from weeks to months, business decisions were often delayed or based on inadequate information. As personal computers, networked workstations and software technology evolved, the mainframe was not the only computing environment. With the development of multiple computing technology, system compatibility and communicability became problems.

Eventually, the University's Vice-President for Business Services hired a consultant who clearly indicated that a new direction was needed. This led to *Purdue's Administrative Computing Master Plan (ACMP)* in 1993. Among other findings, this document declared that data belonged to the University, not individual departments. It set a goal of making data directly available to the person needing it. ACMP recommended that Client-Server architecture be used to meet these goals.

In 1994, a MI led project was undertaken to provide easier, direct access to administrative data for the University's business services staff. In a nutshell, the project developed a Financial/Employee/Student data warehouse running Oracle® on Hewlett-Packard HP/UX computers. Brio Technology's BrioQuery™ was selected as the data-querying tool. Decisions on the hardware, & database infrastructure were made without user input. Users had significant input in the selection of BrioQuery.

Today, the Financial/Employee/Student data warehouse is a very valuable tool for its intended audience, i.e. the business administrators and Student Advisors of the University. However because of the warehouse's monthly refresh schedule, it is less useful for the departmental information analyst who requires not only access to more current data, but also frequently a 5 to 10 year span of historic data.

On numerous occasions customers ask questions which cannot be answered by the Decision Support System (DSS) and/or Brio Query. These limitations include:

- Specific information is not found in the data models.
- DSS data is not current or unvalidated
- Data definitions are different than how DSS defines it.
- Criteria cannot be set on pivot table and detail report numeric totals.
- Customized data models are difficult to create.
- Local user databases and legacy system data cannot be easily combined with DSS data.
- Sophisticated Brio reports are difficult to create.

PROPOSAL TO BRING IN SAS/CONNECT

In 1996 those who attended SUGI 21 in Chicago saw the advantages of the PC/SAS and SAS/CONNECT system. The users initiated a proposal to license the SAS/CONNECT product. Due to other MI projects, the initial proposal was not accepted due to limited and overextended resources.

Users, convinced of the value of SAS/CONNECT, continued to press MI for approval to license and implement SAS/CONNECT. In mid 1997, following the process of building the warehouse, planned phase out of other mainframe reporting tools and the long-term enlightenment provided by ACMP, it became clear to MI that departmental analyst's needed another data access tool.

Project Deliverables

The following product deliverables were identified in the proposal:

- Identify SAS as the tool of choice for power users.
- Upgrade the mainframe and PC platforms with the current versions of the SAS software.
- Reestablish the MI resources that were once devoted to SAS. These include knowledgeable help desk personnel and a training program.
- Installation of the SAS/CONNECT product to permit users of PC workstations to directly access mainframe data.

Project Benefits

The expected benefits of the project included:

- Provide a solution to the limitations of DSS and Brio Query. For example, users will be able to perform sophisticated statistical analysis, include data fields not in DSS, set criteria on pivot table totals, and generate multi-year historical comparisons.
- GUI interface provides a convenient method for Power users to access data.
- Provide seamless interface between mainframe legacy files, Oracle (DSS) database tables, and local user files. Users will be able to concurrently use multiple databases from different sources.
- Perform data processing on the most appropriate platform. Users will have the flexibility to process reports either on the mainframe, LAN, data servers, UNIX boxes, or local PC. However, network resources must be considered.

- Leverage existing SAS knowledge. A large amount of the University's critical information is generated using SAS programs in Internal Audit, Budget and Fiscal Planning, Registrar, University Development, Financial Aid, Affirmative Action, Admissions, and Personnel Service. SAS users have a large knowledge base to build on; thus, fewer resources would be needed to adopt a power users tool.
- Facilitate extraction of mainframe data and transport to the PC workstation where the complete suite of SAS products was licensed. Provide enhanced statistical analysis such as linear and non-linear regression, general linear modeling, multivariate analysis, time series analysis, discriminate analysis, factor analysis, neural network applications, and project management functions. These features will allow for increased accuracy in data analysis and manipulation.
- Begin using additional functions of the SAS product suite. SAS offers enormous capabilities and functions that should be exploited.

Project Approach

Upon project approval by the Department Computing Managers a project team was formed. This project team consisted of one MI staff member and selected individuals from respective departments. Based upon conversations with SAS Institute and MI staff, implementation of SAS/CONNECT was expected to be straightforward.

Administrative processes were developed for granting TSO accounts to departmental users, since these were not historically available to departmental users.

Management Information staff was responsible for the server side installation and customization of SAS/CONNECT while the departmental user dealt with the client-side implementation and testing the interconnection.

A subproject was designated to research and evaluate training alternatives, which included purchased CBT's, external training resources, and development of in-house training materials using SAS SelectText. SelectText is a service of SAS Publications Division, which provides customized SAS textbooks for training.

Another phase was to assure that the most users had access to the most current version of PC/SAS. This required coordination with several different technical support groups across campus.

Finally, continued SAS support is necessary. A centralized resource area should be implemented to work in conjunction with the existing SAS user self-reliance environment. These Management Information staff will work with SAS users to maintain the most up-to-date SAS environment and assist in problem resolution that cannot be resolved via the SAS users.

IMPLEMENTATION OF SAS CONNECT

SAS/CONNECT is basically client-server architecture. Thus there are two components to be implemented. In this situation, the IBM mainframe is the server, and the PC workstations are the clients.

Purdue University has a SAS site license for personal computers using the Windows 95 and Windows NT operating systems. This license includes SAS/CONNECT. The first step of the implementation project was to ensure that all participating staff had the current version of SAS (version 6.12) installed their workstations. Since we have

had over 5 year experience with PC/SAS, this step caused no problems.

At the time this project was started, we were running SAS ver 6.08 under IBM's MVS operating system. The next step in implementation was to install SAS 6.09 under both MVS and TSO on our IBM 390-mainframe computer. Despite minimal previous TSO experience, this step also was completed without difficulty. Testing of this version of SAS was done in both MVS, and native TSO prior to attempting to establish a SAS/CONNECT session.

For our purposes, TCP/IP was selected as the communications protocol to be used between client and server. The generic CLIST supplied by SAS Institutes was modified per the SAS/CONNECT installation guidelines to identify the location of the necessary TCP/IP resources.

The final two steps in the initial implementation phase were customizations needed on the client side.

First we needed to understand and set the value of several parameters on the PC. These are outlined below, and we chose to set them by means of commands that are executed by the PC's AUTOEXEC.SAS file.

Of the several possible communications protocols that can be used for linking Windows/Windows NT clients to a MVS server, we used TCP/IP. This option is set with the following command:

```
OPTIONS COMAMID=TCP;
```

Next we identify the remote host we wished to connect to. This is done by assigning the internet name of the computer to a macro variable, and using that value to set the REMOTE option as follows:

```
%LET rhost=your.remotehosts.name.here;
OPTIONS REMOTE=rhost;
```

Finally we need to identify the connect script used for controlling communications with the remote computer. This is done by setting RLINK, a reserved file reference to the complete file specification of your customized connect script file.

```
FILENAME RLINK 'C:\SAS\CONNECT\SASLINK\script.fil';
```

The second and final PC side customizations were to the "connect" script used to initiate the communications link to the server.

Because we were using TCP/IP and TSO, we started with the generic "TCPTSO.scr" script file provided by SAS Institute. There are two aspects to customizing the generic connect script. The first is to identify the CLIST used to start SAS on the TSO machine and to identify the version of TCP/IP running on that machine. Both values are entered on the same line in the script file and are clearly identified in the customization guidelines included in the comments of "TCPTSO.scr".

The second aspect of modifying the generic script file concerns how the client PC interacts with TSO prior to invoking the CLIST to establish the SAS session. These interactions include not only such routine activities as prompting for user id's and passwords, but also situations such as mis-keying an id or password, and how to deal with an expired password. We initially worked by trial and error

to establish a functioning SAS/CONNECT session. Later as we learned more about TSO, we made additional modifications to produce a more robust script file.

TRAINING

As the technical phase of implementation was completed, it was clear that SAS users who were proficient in the batch environment needed a new mindset to program effectively in an interactive environment.

Background

One key principle to obtaining user acceptance of new tools is to assist the user in feeling comfortable with using the tool. Generally, it is not a good management practice to expect staff to use new tools without adequate training (although we all too often do this - only to see productivity drop rather than increase). Our team kept track of skills used during implementation and used this knowledge to develop training to get other users comfortable with the tools.

Since most of our users came from the IBM mainframe and RJE environment, we started with an introduction to PC/SAS, particularly SAS Display Manager System (DM). This was done so staff could enter, edit, and submit SAS programs. We taught the initial skills, then had staff do a couple of exercises to reinforce learning.

Once basic PC/SAS was covered, we discussed changes to the PC/SAS configuration to enable SAS/CONNECT to function in our environment (Windows95/NT and IBM MVS). This was followed by an introduction to the new SAS commands that allowed for remote file access, remote processing, and the transfer of data between systems. Again, a couple of exercises were used to reinforce these concepts.

We gave the class a couple of weeks to use the skills taught, then offered a structured laboratory where they could work on projects and get personal assistance from the implementation team.

The second key principle to obtaining user acceptance of change is to let the user know that there is an ongoing support system available when the user runs into difficulty or is stumped on how to accomplish a task. We have provided a four pronged approach to ongoing assistance, in addition to RTFM ("read the fine manuals"). We first provided an administrative user's listserv for users, within Purdue, to assist one another, which functions very much like the international listserv - SAS-L. Second, we developed and are constantly improving a web page [<http://www.adpc.purdue.edu/SAS>] with our local information on it. We use the web page to disseminate information about our user group and about local SAS information - common shared macros, formats, etc. Third the "power users" group will provide limited assistance to users to develop concepts into working code. Lastly, we have established an administrative SAS user group on campus that meets quarterly to discuss topics of general interest. One of our "power users" facilitates these sessions. Presentations are frequently from the user community - staff sharing how they use the SAS system in their work.

Additionally, some of our staff has taken advantage of the formal SAS training classes.

We believe that the commitment to internal training for staff and an ongoing support structure are essential to the successful implementation of SAS as an ad-hoc tool in the workplace.

Fortunately, we have convinced our administrative computing center of this and they are committed to providing ongoing initial training for new staff; and to provide technical support over the long term. Unfortunately, most of this support staff is starting from ground zero and has to learn SAS from the beginning, so it will be a while before they can fully assume this role.

Identifying differing user populations and their training needs

As the core implementation team considered wider implementation for the PC/SAS system and SAS/CONNECT, we determined that there were three user groups that we should plan to support with training. Our Administrative Computing Center has established a time line for removal of RAMIS as an ad-hoc reporting tool. RAMIS is to be removed by 12/31/2001. We know that there are several thousand RAMIS programs in production system and in many users' libraries. Some of the users have been using both SAS and RAMIS and some of the users were using RAMIS exclusively. The third group of users merely submitted RAMIS programs that others had written. We started out identifying our population needing training by analyzing the IBM MVS System Management Facility logs. Fortunately we use Merrill's MXG System for Capacity Planning (a SAS tool which is used to extract SMF data and put it into SAS datasets), so obtaining information on which users ran RAMIS and SAS programs was fairly easy to obtain. Once we identified our population, we mailed some questionnaires to measure the skills of these staff members.

Our analysis of data for the period April 6, 1998 through July 6, 1998 showed that we had:

- 190 users who use RAMIS exclusively
- 52 users who use SAS exclusively
- 45 users who use both RAMIS and SAS

We planed to develop three training programs. First to be addressed were the staff who would have to convert RAMIS programs to SAS. Then we would plan to develop initial training for new staff in SAS. This training was going to include knowledge of the legacy data and systems as well as tool training for SAS. Lastly, we would consider training for existing SAS users to widen their skills within the SAS system.

Currently, work is underway, in cooperation with the SAS Institute Publications Division, to develop training materials. We are taking advantage the SelecText service to combine pages from regular SAS manuals as well as SAS training manuals. The Publications Division has told us that this is the first time they have included SAS training materials in their SelecText offerings.

Basic Windows

While most of our users have desktop computers that use the Microsoft Windows 95/NT operating systems, we found that some users have trouble starting SAS and in configuring the desktop for efficiency.

We found we had to provide guidance in locating the SAS icon to start a PC/SAS session. If one uses the standard PC/SAS install the program can be found in the path "Start -> Programs -> SAS -> SAS".

We also showed the class how to quickly create a desktop icon so they could access the SAS system quickly (right click on the desktop background, New -> Link -> Browse -> C:\Program Files\SAS\SAS.EXE, enter, "SAS", enter).

Instruction in PC/SAS

Our initial class of users, who learned about SAS/CONNECT, was mostly experienced SAS programmers who worked in the IBM-MVS mainframe environment using a mainframe editor and job submission facility (ROSCOE). They were familiar with the edit/run development cycle of batch processing, where the entire job was processed with each job submission.

We believed it was important for the users to understand the interactive development cycle where processing is performed a step at a time, and analyze data at each step, until you have obtained the desired results. It was also important for the users to understand that PC/SAS session is "persistent" - that is, all work you do stays around until you explicitly release it or close the session. In batch, options and work datasets only exist for the duration of the job. In interactive SAS, datasets and options hang around for the duration of the session. This can lead to unexpected results sometimes.

Display Manager Commands

Once the SAS session is started on the PC, the user is greeted with the Display Manager (DM). DM is the Microsoft Windows application program interface to the SAS system. DM includes a program editor, windows for SAS output, the SAS log, and commands for managing the session. The menu bar changes depending on which window you are in (context sensitive).

Program development is started in the program editor. The program editor is a very powerful text editor. Within the editor you can open exiting SAS Programs, save new SAS programs, and recall previously submitted SAS code. The editor supports the standard Windows editing (cut, copy, paste) as well as line commands (insert, delete, delete block, move, move block, replicate, replicate block, etc.).

For those familiar with other integrated programming environments (such as Microsoft Visual C++, Borland Turbo C++, Borland Delphi, Microsoft Visual Basic, Microsoft J++ Builder, Symantec Café) the PC/SAS system will appear familiar.

Instruction on SAS/CONNECT script

Our initial training on SAS/CONNECT included a walk through of our local adaptation of the standard SAS/CONNECT script for TCP/IP communications to TSO on our IBM-MVS/ES from the PC. While we didn't expect users to be experts at script development - we did think it was helpful for them to see and understand the script. If they had problems later on, it would be helpful for them to understand the concepts in the script.

We also made our standard script available on our SAS Web Site so users could obtain a copy easily and install it - without having to key in each command line.

Details for programming a SAS/CONNECT script can be found in Chapter 4 of the SAS/CONNECT Software: Usage and Reference, Version 6, Second Edition manual.

Instruction on AUTOEXEC.SAS and CONFIG.SAS configuration files

The SAS System uses two files to provide user customization of the SAS session. The CONFIG.SAS file is read at startup and sets options for the SAS environment. The AUTOEXEC.SAS file is read and processed at the end of the startup process, each command in the AUTOEXEC.SAS file is executed in order - just as if the command had been typed into the immediate execute box.

By utilizing the AUTOEXEC.SAS file, we were able to show users how to invoke a SAS/CONNECT session and then established links to remote permanent SAS datasets and external files by means of LIBNAME and FILENAME statements.

Additional information can be obtained on the CONFIG.SAS and AUTOEXEC.SAS files by referencing the SAS Companion for the Microsoft Windows Environment; Version 6, First Edition in Chapter 1, "Files Used by the SAS System".

SAS/CONNECT Training

During testing the implementation team quickly learned the basic commands needed to manage the SAS connect sessions. SIGNON and SIGNOFF are two obvious commands that initiate and terminate the SAS connect session to the remote host. PROC UPLOAD and PROC DOWNLOAD can be used to transfer SAS datasets or external files to and from the remote machine. RSUBMIT is used to send SAS program code to the remote server.

In the training session the syntax and purpose of these commands was quickly covered. However, the concept that the programmer/user is controlling two SAS sessions (local and remote) required much more time and explanation. The local session is functioning in front of him on the PC and the remote session is on the mainframe. We could not over stress the importance of this concept.

Failure to keep the two sessions in mind can result in several common errors.

- Submitting program code to the wrong processor
- Setting options or macro variables on one processor and expecting them to be in effect on the other
- Attempting to assign a libref or fileref on one platform when the files exist on the other

In our experience, these errors do not cause catastrophic. However, losing track of these two SAS session can cause you to look at syntactically perfect SAS code and wonder why you did not get the expected results.

Typical Program Development and Debugging Cycles

We explained the differences between the "batch" processing cycle as we see in Remote Job Entry and the "interactive" processing cycle as we see in PC/SAS.

In the batch processing cycle, one alternates between an editor tool and running the job. Output from the job is displayed. The log is checked for warning and error messages, and the record counts are examined for reasonableness. Output files are checked for proper results - content as well as presentation. If there are errors, warnings or cosmetic changes to be made, we slip back over to the editor to make these changes. Each time we are changing the tool we use. In the batch cycle, one may use two, three or more different tools to accomplish the task (edit, submit jobs, track running jobs, display output, and route output to print).

Frequently large processing jobs are developed a step at a time. The initial input section may be developed with a simple print of a sample of the data. The second run may add some more processing steps with prints to examine intermediate results. After several cycles the task will be completed to satisfaction. Each batch submission involved rerunning all the preceding code - which was wasteful of computing resources. If one saves intermediate processing results, you get into data management issues - keeping the data fresh, backed up, and so forth. In addition, with intermediate saves, you have to break the processing down into steps, which later have to be changed to run as one procedure.

In the Interactive processing cycle, you accomplish one step at a time also - but all of your work is saved in working buffers, and your SAS datasets remain in existence until you explicitly remove them or close your SAS session. This means that the data step can be processed, and then you can examine data (using FSVIEW - an interactive SAS data set viewer - providing a view of SAS data sets on the screen), and then you can move right on to processing the data - the data step is only run once. Your programs are held by the PC/SAS system in program buffers and can be recalled at any time to build processing steps, or to edit code with errors or warnings, or to change data presentation or analysis. Furthermore, with interactive processing you can take a variety of looks at the data for trends, patterns or other facts that you might not do with the batch system. The interactive means of program development offer the programmer great flexibility.

The advantages of working in an integrated programming environment have been realized by the major publishers of development tools - Microsoft, Borland, Oracle and others provide tools which work in this integrated interactive development environment to improve programmer productivity.

Hands on Sessions

The implementation group felt very strongly that trainees assimilate information more thoroughly when they perform hands on exercises. We also felt it was important to provide positive reinforcement of the concepts taught. We provided four exercises during the formal training - two in local mode, and two using SAS/CONNECT. In addition, we provided an "open lab" about two weeks after the formal training where staff could bring in their projects for "consulting" assistance.

The formal hands on sessions involved very simple exercises that included reading external data and doing simple data manipulation (print, sort, means).

The exercise data, and the sample solution programs were posted to our local SAS Web page so the trainees could work further with the exercises in their offices if they wished.

Small test datasets (~ 25 obs.) were used to maximize learning time and minimize processing time.

During the hands on sessions the "power users" circulated around the room and made contact with each person to assure they understood and were able to use the lecture material. Individual supplemental instruction was provided as needed.

While we make knowledge of the SAS language a prerequisite for the course, we found that the users in the class had a variety of skill levels.

Check on New Users

At the end of our formal training session we had each participant fill out a training evaluation form. This feedback helped us to analyze the presentation and determine how we could improve the class the next time it is delivered. We realized in the training that we used a large portion of the time covering basics, which results in our time for the hands on demonstration of SAS/CONNECT to be cut short.

We also planned for, and used a full afternoon for this training. We should be more efficient in the use of the time allocated, and realize the work loads of staff - ideally training should be held to two hour formal sessions.

We realized the diversity of the skills of the class when we read the evaluations. The comments ranged from "to complex - the class moved too fast" to "to simple - the class moved slowly". From this we identified that separate training is needed in basic skills - using Windows, using PC/SAS, etc. Users should have a strong working knowledge of PC/SAS before moving on to the SAS/CONNECT environment. Some of the users were confused with the client/server environment - not understanding the concept of distributed computing.

These comments are helpful in assuring the staff understand comments. If you provide training, or oversee training, you should use this feedback to assure the training is relevant.

Bring Back Real World Examples for Users to Continue Education

We gave the users about two weeks to work with the initial training, and then we scheduled an "open lab" where users could bring in their "real life / work related" problems for technical consultations with the "power users".

Approximately six of the original group of nineteen took advantage of this opportunity.

Establish Sources of Additional Help (Local to Purdue)

We believe strongly that a successful integration of SAS into the workplace as a routine tool depends heavily on (1) initial training, and (2) ongoing support.

We believe users should start with self-help by "RTFM" ("read the fine SAS manuals"). We also realize that to use the manuals, one has to have an idea of where to start. The massive SAS documentation can be overwhelming when

getting started. We have provided alternatives for the users to pursue.

First, we prod an administrative users listserv for users within Purdue to assist one another [this functions very much like the international listserv - SAS-L].

Second, we developed, and are constantly improving, a local web page [<http://www.adpc.purdue.edu/SAS>] with our local information on it. We use the web page to disseminate information about our user group and about local SAS information - common shared macros, formats, etc.

Third the "power users" group will provide limited assistance to users to develop concepts into working code.

Fourth, we have established an administrative SAS user group on campus that meets quarterly to discuss topics of general interest. One of our "power users" facilitates these sessions. Presentations are frequently from the user community - staff sharing how they use the SAS system in their work.

Register Users with SAS User Group

The local SAS listserv is used to communicate information about our User Group meetings - schedule, agenda, etc. As new users are identified (or upon an individuals request) they are added to the listserv distribution list.

We also use the listserv to publicize the Midwest SAS User Group meetings and the International SAS User Group meetings. We coordinate attendance and transportation to these conferences using this facility.

Our local Purdue Administrative SAS User Group has applied for and been recognized by SAS as an official user group with SAS Institute support.

Web Page Established to Assist Users

As part of our ongoing user support, the pilot group acquired space on the Administrative Web Server for web pages about "SAS and our local specific information.

We stored configuration files on the web page which can be accessed and stored on the local computer.

We've stored sample test data, exercises, and solutions on the web to facilitate training sessions.

We started building up local documentation on the web page - local macros we've developed, how to use COB2SAS (the SAS free utility that converts COBOL copy books to SAS Data steps), stored formats, etc. We have developed a hierarchy of shared facilities - a core set of facilities are in common files which are attached to every SAS session by default. The web page is our method to document these command-shared facilities.

We also use the web page to communicate schedules for our Administrative Computing SAS User Group Meetings.

The web page is intended to be our "one stop" location for Administrative Computing SAS Users.

Department contacts are established to assist users

We identified SAS users with areas of the University who can "champion our cause" and provide support to their peers. As mentioned previously, we analyzed computer use data to identify SAS users. We used this information to build up a network of staff who can assist new users. Our central support then provides support for these departmental contacts. The departmental staff provided close and quick access to assistance for problem solving.

University List Server established

Purdue University Computing Center (Academic Computing) runs listserv software to provide mailing list services for the campus. We registered the list "PUSASU-L" - Purdue University SAS Users List on the listserv and add people interested in Administrative Uses of SAS at Purdue.

The listserv is an open forum for discussing questions or problems with SAS within the Administrative Users at Purdue. In this regard, it functions much as SAS-L does for the international SAS community, but only for Purdue Administrative users.

We disseminate information about our local, Midwest region, and international SAS conferences to the list.

The list operates as a closed list to minimize the opportunity for mail spamming (mass delivery of unrelated or undesirable solicitations). Members can only be added by request to the list owners. Only members of the list can post to the list.

CONCLUSION

In conclusion, current trends in electronic information have dictated that Purdue University should begin to utilize other products and services to meet the ever-increasing needs of its users. Recent technological advances have raised the demands and expectations of data providers. As it becomes evident that customers are better educated in the information/data products and services available, the responsibilities and daily tasks of service providers have dramatically changed over the past few years. Users now have a constant need to analyze and manipulate data that is not readily available on their desktop workstations. Users frequently request data from the mainframe to be manipulated on local workstations using, for instance, Microsoft Access[®]. In conjunction with this change, data providers are now expected to query data on multiple operating systems. This change has necessitated a need for the "power" user to query data on multiple platforms with a minimum of time allowed.

With the introduction of SAS/CONNECT, the "power" user now has the capability to query data, quickly and efficiently, from the mainframe. These were the trials and tribulations associated with the installation of SAS/CONNECT in a computing environment that had become stagnant over the years.

REFERENCES

SAS Institute Inc. (1996), *SAS Companion for the Microsoft Windows Environment; Version 6, First Edition*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1994), *SAS/CONNECT Software: Usage and Reference, Version 6, Second Edition*, Cary, NC: SAS Institute Inc.

ACKNOWLEDGMENTS

The following staff members from Purdue University contributed to the preparation of this paper:

Nathan L. Bohlmann W. Mike Doyle

The following staff members from Purdue University were on the implementation team:

Joseph M. Braun W. Mike Doyle
 Micheal S. Hines C. Kee Lee
 Mindy Wang Shawn L. Williams

SAS and SAS/CONNECT are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Oracle is a registered trademark of Oracle Corporation.

IBM is a registered trademark of International Business Machines Corporation.

Other brand and product names are registered trademarks or trademarks of their respective companies.

Authors:

Joseph M. Braun
 Purdue University
 1126 Freehafer Hall
 401 S. Grant Street
 West Lafayette, IN 47907-1126
 Phone: (765) 494-7412
 Fax: (765) 494-6138
 E-mail: joebraun@purdue.edu
 (10+ years SAS experience)

Michael S. Hines
 Purdue University
 1280 Engineering Administration
 West Lafayette, IN 47907-1280
 Phone: (765) 494-5338
 E-Mail: mshines@purdue.edu
 (10+ years SAS experience)

Shawn L. Williams
 Purdue University
 1095 Hovde Hall
 West Lafayette, IN 47907-1095
 Phone: (765) 494-6148
 E-Mail: smwilliams@reg.purdue.edu
 (2 years SAS experience)