Supporting a Global SAS® Programming Environment? Real World
Applications in an Outsourcing Model
Karen Curran and Mark Penniston,
Omnicare Clinical Research, King of Prussia, PA

ABSTRACT
Talented individuals knowing SAS in the Pharmaceutical Industry are sought-after gems. One advantage to globalization is to tap into those markets where superior SAS programmers and statisticians can be trained to bolster the size of the group. Our experiences in providing worldwide, 24-hour support in 6 geographic regions have led to some interesting solutions both from a systems architecture basis and an implementation model. Integration Technologies, Enterprise Guide, and a CDISC platform have helped in creating an agreed-upon vocabulary among the various sites, but without other skills and tools there would not be a truly standardized programming environment. As early adapters for all three technologies, this paper presents our experience with implementing, supporting, communicating and upgrading a worldwide SAS programming environment and set of processes.

INTRODUCTION
This paper describes the thought process behind our development of a global SAS programming platform with use initially occurring in the U.S. but looking to the future with multiple international sites being initiated as company process harmonization and growth occur. It then describes the practical points that happened as the environment was used and the incorporation of CDISC as a global programming standard has been implemented within a CRO environment with multiple customers, each having their own standards.

DESIGN AND IMPLEMENTATION OF A GLOBAL ENVIRONMENT
The genesis for the system architecture was derived from the SAS programming population residing within the company. The populations were those of:

- Individuals accustomed to VMS and a command prompt
- Individuals accustomed to a PC SAS interface
- Individuals accustomed to Unix style editors

In addition, the programming environment required:

- Directory and file naming conventions with rules
- A forward looking method to communicate with Web based platforms
- Reproducibility and audit trail methodology
Figure 1 presents the initial physical architecture of the programming system.

The decision was made to use a centralized server based on the experience of harmonizing multiple servers for SAS processing. We implemented first with only one site and progressed to others as project demand dictated. Validated program files would not require duplication (i.e. one global validated SAS Macro library) and team members from multiple geographic locations could access the same files and output, preventing management from worrying about duplication of work. Using SAS Integration Technologies and Enterprise Guide we were able to present a solution that those accustomed to working under a PC SAS platform would be comfortable with. Those users more accustomed to a Unix or command style environment used terminal emulators directly to the Unix platform editing the SAS programs on-line.

Every programming environment on the production machine consisted of a development and production area for SAS processing to occur. Version control software was installed on the production machine such that a history of all changes would be recorded for audit purposes as programs were promoted from development to production. Stored within the version control database are such items as who last updated the production code, what was changed and the reason why the program was placed there. Users were at first uncomfortable with the idea of checking in code but now appreciate the tracking of changes with historical documentation.

To ensure a person is only working on a single project setup, the Unix directory structure was created so that when working in one project a user cannot be in a second, unrelated project at the same time. Unix scripts were created and validated for directory structure configuration and accessing of files. Batch processing of the SAS programs could only be achieved by one first registering with that project.

Example UNIX command registry to a project.

```
OCRSAS% register myclient project
Registering to task “myclient project displays dev”
OCRSAS%
```
With Enterprise Guide this registration is accomplished before a user enters the system. A customer/project selection routine was designed and written to provide a pick list from which the desired project could be chosen. Once all options were selected, Enterprise Guide would then start up initializing to the appropriate customer specific macros for proper functionality.

Example Enterprise Guide registry to a project

In both instances a user can define if they are entering the development or production environment. Additionally, one is allowed to choose which version of Enterprise Guide to work with as there are certain interface characteristics that users prefer. The version of SAS that can be used for that project is stored within a particular projects information summary repository and is not an accessible item from either the interactive or command prompt project entry utility.

Platform implementation occurred first in the U.S. sites (Pennsylvania and New York) followed by an international rollout (Germany and United Kingdom). It was believed that this staggered approach would help in identifying and correcting of unforeseen problems before release to a wider audience. There was no intention of migrating existing projects from other platforms to the new system which would allow us to have gradual growth of the platform and to monitor system performance. In retrospect, this proved to be an ideal situation for managing growth of the system and user acceptance for the new platform. Dataset harmonization was agreed to in 2003 with the publication of more defined CDISC standards. It was believed that as we moved toward harmonized default output, the data from which said output was based upon should be harmonized too.

Initial Worldwide Implementation
SUPPORTING AND UPGRADING A GLOBAL ENVIRONMENT

The support of a global environment requires:

- Training
- Documented processes
- Communication
- Feedback and review of performance metrics
- Constant reexamination

Documentation of processes was performed through updates of Standard Operating Procedures and Working Practice Guidelines. This was supplemented with training and multiple communications to the user community. Feedback was always blunt and to the point from this particular community with the adverse effect of delayed acceptance toward migration to the new platform. The concept of using both a development and production environment separated by a version control software system caused a fair amount of communication within the community which had, to that point, not fully accepted the concept of code validation. Standard datasets based on CDISC definitions were met with skepticism until the community began seeing programming efficiencies. This was true both from an internal and external client base.

Implementation Findings:

- New systems should not be slow.
- Determining why they are slow can be a difficult process.
- Agreement to database and output harmonization can be a difficult but very rewarding process

Through methods of deduction we have discovered that:

- Enterprise Guide works better with DCOM servers assisting in the location of the SAS user community
- Network speeds are dependent upon internal, in-house hardware as well as external service provider speeds.
- Above a certain distance, desktop virtualization software such as CITRIX must be used even for terminal emulation software to be usable
- Everyone agrees that standards are required to gain the ability of sharing resources globally
- Adding additional sites to the platform, requires training and a high degree of project sharing to ensure all operational definitions are harmonized.

Unix with Integration Technologies and SAS Enterprise Guide appear to communicate to one another more than initially thought. Where it was originally believed that EG sessions only required the information of where the SAS machine is located and what port to use, it appears that there are communications regarding binders, groups and folders on an ongoing basis during the applications execution. To assist in this communication process a series of local DCOM (distributed component object model) services were placed in each of the 4 offices. This increased the speed for an EG session tremendously.

There was a large investment to increase the bandwidth of the remote offices to the central server to obtain acceptable performance standards for network speed. We discovered that one office (Germany), while having the same network connection as the United Kingdom office, could not achieve the same relative performance. It was discovered through a manual process that circuitry and hardware within this office was faulty, requiring upgrade or replacement. These problems were not discovered prior to this implementation as the SAS server and data were local to that office until this upgrade. Having the old system remain available for processing of data allowed for such findings to be worked through without impacting project work.

As the company has since grown we have found that above a certain, undetermined, distance from the central server there is such a degradation of performance with Enterprise Guide or terminal emulation that programming cannot realistically be performed. To handle this we have since turned to using desktop virtualization software to get us to acceptable performance levels.
The figure above graphically displays the current worldwide SAS implementation. It allows the user community the greatest flexibility feasible in a central server system without implementation of Web based SAS tools.

In parallel to the global architecture implementation, global process harmonization has been implemented as well. Written in SOPs and Working Practice Guidelines the processes are described to ensure conformity as best as possible. The balance of harmonization for efficiencies to the creation of bureaucracy is a serious concern within the organization. To that effect we choose CDISC as the standard data model to remove our internal conversations and to help new individuals to the company. Even with published standards there are unanticipated interpretation discussions that occurred. We have discovered that only by working together on projects are we able to completely ensure the primary vocabularies between groups are the same. Once a few projects are shared together the project teams become site based for further efficiencies in programming production.

CONCLUSIONS
Planning of Global systems is only the first step in harmonization activities. Implementation of systems requires constant communication and rethinking of initial assumptions when the SAS software is implemented in a version 1 or 2 form. SAS is an excellent tool to implement such efforts. The ability to take internal customer feedback about a system is just as important as initial rollout for user acceptance to occur. Only through user acceptance will the benefits of such systems be achieved and efficiencies realized.

REFERENCES:
ACKNOWLEDGMENTS
SAS is a Registered Trademark of the SAS Institute, Inc. of Cary, North Carolina.
CDISC is a non profit organization dedicated to develop and support global, platform-independent data standards that enable information system interoperability to improve medical research and related areas of healthcare.
Special thanks to Michelle Hayden for all her help in putting this document together.

CONTACT INFORMATION
In case a reader wants to get in touch with you, please put your contact information at the end of the paper.)
Your comments and questions are valued and encouraged. Contact the author at:

Karen Curran and Mark Penniston
Omnicare Clinical Research
630 Allendale Road
King of Prussia, PA 19406
Work Phone: (484) 679 2419 (484) 679 2436
Fax: (484) 679 2509
Email: Karen.Curran@omnicarecr.com
Mark.Penniston@omnicarecr.com
Web: www.omnicarecr.com

*******************************************************************************