

## Statistical Computing Environment Implementation – An Agile Approach

Gary Cozzolino, d-Wise Technologies, Morrisville, NC

### ABSTRACT

Historically, companies in the pharmaceutical industry have stored SAS<sup>®</sup> data, programs, and output on Windows or UNIX file systems, and executed SAS on shared servers or locally on PCs. More organizations are now moving to “Statistical Computing Environments” (SCEs), which provide secure, audit-trailed computing platforms, enabling programmers to effectively manage their programs through the Software Development Lifecycle. As more companies move to SCEs, pharmaceutical, biotech, and CRO organizations face a number of challenges in implementing such systems. Regardless of the technology involved, and whether a commercial product is used or if it’s a home grown system, significant changes to related business processes should be expected. Among these challenges, programmers will need to learn new processes for program development, management and execution. This paper outlines an agile approach to implementing these enterprise solutions. The iterative methodology begins with identification of priority business processes, followed by rapid piloting and revisions, leading to development of formal processes for production. Other key aspects to be considered include change management, training, and study migration to the new environment.

### INTRODUCTION

Implementation of an SCE is an activity that has far-reaching impact on the entire clinical trial analysis and reporting process, affecting business processes in clinical data management, statistical programming, and biostatistics. While SCE technology may vary widely, from commercial products such as SAS Drug Development or entimICE<sup>®</sup>, to simpler systems utilizing a SAS server with a file version control system, successful implementation depends not only on the underlying technology, but equally (or arguably - even more so) on the related business processes and associated organizational change management efforts. Having participated in a number of SCE implementations, d-Wise has developed a methodology that enables organizations to consider both the process and technology together, maximizing the SCE value proposition. This paper will outline the methodology, which entails piloting and configuration of a Development or QA environment. Full system validation and production system configuration and rollout are separate activities but will be briefly discussed.

### APPROACH

Organizations often approach system implementation using a linear methodology where user and system requirements are developed, the system configured, validated, and released for usage. We recommend usage of a phased approach where the technology and process aspects are considered together in development of a strategic implementation plan. An “SCE Pilot” will be conducted to allow critical workflows to be developed, prototyped, and tested. From this iterative pilot project, formal business processes are developed and SCE configuration settings are determined. Upon completion of the pilot, steps toward system qualification and planned production rollout will follow.

In conjunction with the pilot, change management activities should be taking place – easing the transition for the user community. Also, efforts related to integrated user training and study migration must be considered.

It is highly recommended that at least two distinct SCE environments be set up as part of the implementation process: a Development environment, where the pilot will be conducted, and a Production environment, where the final, validated instance of the SCE will reside. With fewer controls required, the Development environment allows the team to quickly modify the SCE configuration settings as needed during the prototyping stage – allowing the exploration of “what if” scenarios without any concern for impacting production work. Many sponsors also have a third environment: QA, where system qualification is performed, allowing a “clean” configuration to be tested prior to moving to Production.

## STRATEGIC IMPLEMENTATION PLAN

Before a pilot can begin, it's prudent to have a plan in place to map out the overall SCE implementation, from pilot through validation and production rollout. As this plan is put together, it is also recommended that an assessment is made of the current environment and business processes. This will help ensure that any current gaps and pain points are identified, ensuring that new processes don't repeat past mistakes and those current gaps are addressed as much as possible. At this point, the high level list of priority processes to be addressed in the pilot phase should also be identified. For example, the initial implementation may only include the development of a standard folder hierarchy, user roles and guidelines for moving data and files into and out of the system – using the SCE as a secure repository, deferring SAS code development workflow to be implemented in a follow up phase.

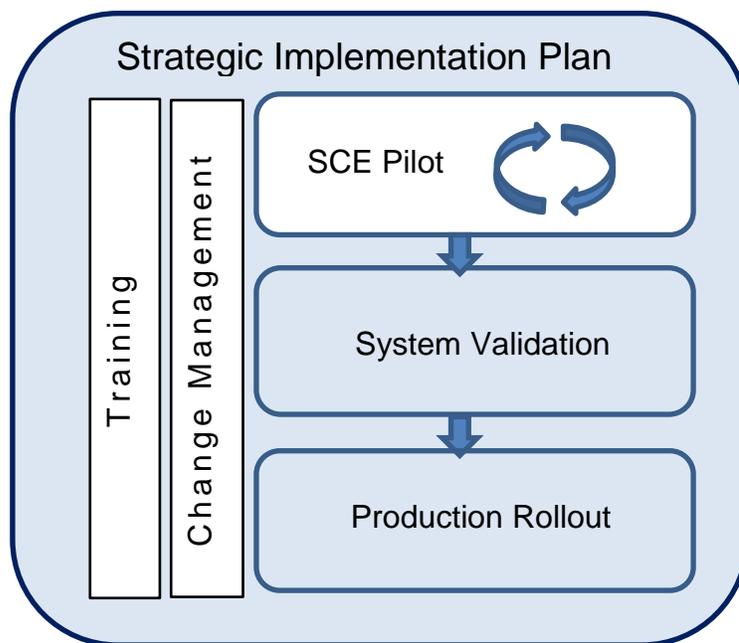
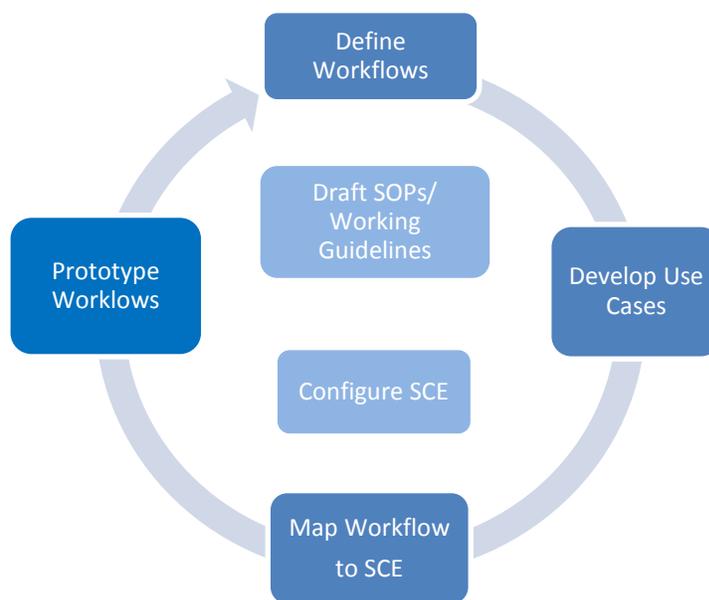


Figure 1. Strategic Implementation Plan

## SCE PILOT

The Pilot phase of the SCE implementation lays the groundwork for the production rollout. Here, we will get real world experience using new processes within the system. A clear feedback loop must be set up to ensure efficient integration of feedback from each iteration. A key component of the pilot is the definition and implementation of business process workflows. In this context, workflows will take the form of flow diagrams that graphically describe a business process, showing relevant user roles and basic steps taken to complete a given task. Having these flows documented visually greatly aids in the information sharing and review processes.



**Figure 2. SCE Pilot Iterative Process**

## **PILOT INITIATION**

The very first step in the pilot is to identify the **Pilot Team** that will work through the pilot process. This team should have cross-functional representation from all groups that will work in the new SCE environment. It is expected that SAS Programmers will be primary users of the system, but it is recommended that other key stakeholders in related business processes are also included. These may be members of the core Pilot Team, or be identified as key representatives for their respective groups that may be called on to participate on an as needed basis. Team members should ideally be future users of the system and/or management of those users. They will also be the SCE Subject Matter Experts (SMEs) in the organization and will be a resource for the larger community.

Prior to initiating the larger process described above, the pilot team will have a project initiation meeting to develop a **project plan** for the SCE Pilot. Suggested content for the SCE Pilot project plan includes:

- Outline of the high level project expectations
- Key objectives, deliverables, and timeline
- Roles and responsibilities
- Communication plan
- Decision-making processes

While the project plan is a key piece in defining “how” the pilot will be conducted, it is also important to ensure that the scope (“what”) is clearly delineated. The basis for the pilot scope will have been identified in the Strategic Implementation Plan, so more detail will be added at this time. It is not envisioned that EVERY possible process be addressed in the pilot phase - this methodology focuses on **key** processes that will need to be defined in the context of the SCE. Lower priority processes may be prototyped and refined at a later stage. The purpose of the pilot is to define how these key tasks will be performed within the SCE, facilitate system qualification and a faster path to implementation of a production environment. It is important for the team to look at their business processes with “fresh eyes”, and not simply fit existing practices into the new system. The team should look to innovate around the processes and leverage new capabilities the SCE brings. In addition to the core SAS programming-related business process priorities, the pilot team must also document the technical scope for the pilot - for instance, will full system integrations (EDC, IVRS, etc.) be an early objective? Or will more manual processes be adequate for initial system implementation? This level of structure is important to facilitate the communication of clear objectives across the project and the organization.

**DO** assemble a pilot team that is forward thinking and open minded regarding their business processes. This group will be instrumental in the delivery of the SCE, and must be committed to its success. It is also critical that project expectations be **clearly** identified in the project plan.

**DON'T** exclude non-core groups from the pilot – all affected groups should be at least made aware of the implementation and pilot plans, if not, directly involved. Also, do not get into problem solving at this time, the goal is to produce the **plan** for the pilot.

## DEFINE WORKFLOWS

Once the project plan is in place, it is necessary to define the high level tasks that will be performed within the system. A **sample** list of key processes to be reviewed:

- User setup and administration
- Study setup
  - Study Team / Access rights assignment
  - Study/Folder creation
- Loading data from external data sources (EDC, Labs, IVRS, PK, CTMS, etc.)
- Study Blinding/Unblinding
- Development of SAS Programs – Development/QA/Production
- Delivery of Tables, Figures and Listings
- Study Archive / Retrieval
- Creation and publishing of data package for FDA submission

Of course, every organization will identify different processes and have different priorities. Once a list of processes (consistent with the scope of the pilot) is identified, it will be prioritized and categorized for inclusion into the SCE Pilot effort. During this stage, skeleton diagrams of the prioritized workflows will be produced. These will describe the business roles and basic tasks involved. It's important to keep these flow diagrams at a relatively high level – getting the group buy-in is key here, as well as ensuring that people don't get caught up in details – they should be designing the optimal process flows that will work best for the organization.

**DO** refer back to the master implementation plan to ensure that the priority workflows are in line with the overall goals and scope. Ensure that the workflows are “system agnostic” – they should represent the ideal workflow, regardless of the technology.

**DON'T** get too detailed at this point – these should be high level workflows that all parties involved can support as the preferred process going forward.

When this step is complete, the team will have created the initial set of **Draft Workflow Diagrams**.

## DEVELOP USE CASES

Once the team has developed the high level workflows, the next step in the process is to design the processes at a more granular level, identifying and defining each step within each workflow. This leads into the development of use cases, which will be the end product of this step. Each use case will identify the relevant business process steps, the user conducting that step, any details the user should take to accomplish the task, and any alternative flows that must be supported.

As with the draft workflows, the use cases should be developed independent of the chosen technologies, allowing the implementation team to focus on developing a workflow that meets their needs and not developing workflows solely driven by the technology. The team should also keep in mind that they will be iterating over these workflows multiple times, so they won't be perfect the first (or maybe even the second) time through. It is important that this expectation be communicated across the team, and that everyone understands that this is a learning process and changes **will** be made over time. An added benefit of this method is that the use cases may later be leveraged for test case development for the system qualification phase.

**DO** choose a use case format that is flexible enough to accommodate more complicated decision trees, but allows easy linking to workflow diagrams.

**DON'T** get caught up making these perfect formal documents – they should be fairly detailed, but know that they will be changing over time as the processes are refined.

**Use Cases** and updated **Workflow Diagrams** are the result of this portion of the process.

## MAP WORKFLOWS TO SCE

After the use cases have been designed, each step in the use case workflow is “mapped” to SCE functionality. What this means is that the use cases are reviewed, and with help from SCE experts (internal personnel, SCE vendor, or a third party), it is determined how the individual steps would actually be performed in the SCE. In some cases, multiple methods will be identified and the Pilot Team will need to determine which option best meets the organization’s needs. In addition to identifying the SCE functionality that is used within the specific use case, documentation needs will be identified. This will help identify where SOPs and working practices will need to be written later within the project.

At this time, the team will also identify where gaps exist between the defined use cases for the SCE and the underlying SCE technology. For example, the use case that covers SAS Program development and validation may describe a workflow where the developer flags a program as “Ready for Validation”, with the desire that the validator be automatically notified. If the SCE system doesn’t have this sort of workflow built in, there may be options to leverage external tools to facilitate this, or the team may opt for a manual process where program status is tracked outside the SCE. The team will determine how to address the identified gaps and whether they will be filled by process, additional tools, or custom code/utilities. The team can then review the planning and development of additional processes and tools for scope, schedule and budget impact to the overall project plan.

**DO** have expert resources available to work on this step – having the proper expertise will not only lead to a quicker mapping of the workflows, but will also ensure that the full SCE functionality is properly leveraged. Without this expertise, there is a risk for the team “spinning their wheels” on technical issues and even worse, not taking advantage of system features.

## CONFIGURE SCE

In parallel to the definition, design, and mapping of the workflows, the team will work to install and configure the development SCE environment to support the Pilot objectives. Configuration settings may include:

- User password requirements (length, special characters, etc.)
- Workflow requirements: Developer must flag program as “Ready for QC” before it can be validated
- Configuration of folders to contain certain file types: Data, SAS Programs, etc.

Of course, system configuration changes must be performed in a controlled fashion, but performing the pilot on a Development environment does allow additional flexibility to make configuration changes more quickly to allow faster prototype completion. This will be an ongoing task, as configuration setting changes may be needed as workflows are prototyped.

SCE configuration options will vary greatly depending on the system itself. A more bare-bones, homegrown system that consists of a SAS server and controlled repository for data and SAS code will have fewer technical configuration options, while commercially available SCEs will have a variety of system options that may be tailored to the organization’s needs.

**DO** plan accordingly to have vendor and/or internal IT resources available as needed to ensure configuration updates are made in a timely fashion and prototyping timelines are not impacted.

**DON’T** have multiple administrators making configuration changes. While these changes will be in a Development environment, a controlled process is still needed around any changes made.

## PROTOTYPE WORKFLOWS

Up to this point, the work has been mostly abstract with little actual work being done within the SCE. It is important that the team NOT get caught up in the design phase too long – we encourage teams to get to prototyping as quickly as possible. After the workflow definition is complete and the SCE is configured and available for conduct of the SCE Pilot, the hands on prototyping may begin. This will lead to refining of the workflows, use cases, and functional specifications as well as a better understanding of gaps and methods for resolving those gaps. Each of these documents will be continuously updated during this process with the goal of delivering robust and complete prioritized workflows. The expectation here is that the team will iterate through this prototyping exercise at least twice and possibly three times during the SCE Pilot.

Prototyping consists of having SCE Pilot members or designees (ideally, future end users of the system), actually perform tasks outlined in the workflows and use cases in the SCE Development environment. Now a fair amount of setup work is expected prior to the prototyping efforts. The SCE Pilot team must ensure that the Development

environment is set up and configured to properly support the users who will be performing the prototyping – this may include:

- All user accounts and groups set up
- Prototypers trained on the system
- Prerequisite test data and programs identified and present in SCE
- External studies/files/systems identified for usage in prototyping
- SMEs identified to support prototypers

We recommend that the prototypes simulate real world experiences as much as possible, but existing code should be leveraged as much as possible – remember, we want to be testing parts of the process that are unique to the SCE to find the best process. It's NOT a time to test user's skill in developing new SAS code. We also recommend that these prototypes be fairly focused efforts, and not stretched out over too long of a timeframe – this would require that the prototypers be dedicated to this task, which can often be an issue for teams, as there are frequently project demands that may take precedence. Plan carefully, and try to minimize this as much as possible – having a focused group that can continuously work through the prototypes is definitely preferred. Do not underestimate the time commitment here, as there is always a learning curve associated with use of a new systems and processes.

**DO** make sure to capture feedback as these prototypers work – feedback on the processes themselves, as well as the system. This information needs to loop back to the team to ensure that the workflows are adjusted accordingly, and is also an excellent source of information for training – identifying areas of concern, FAQs, etc. Encourage prototypers to experiment in the system and trying various methods where possible.

**DON'T** worry about something not working – that's the point of the prototype! Find the weaknesses in the processes and the system and make adjustments. It is expected that the workflows WILL need to change based on the feedback. Also, there may be areas of the use cases that may not be as well defined at the outset – the prototypers will be invaluable to provide ideas and options on defining more precisely what is needed.

## **DRAFT SOPS AND WORKING PRACTICES**

During the entire process outlined above, the pilot team creates and updates SOPs and working guideline documents based on prototype results. These will be initiated as skeleton documents, and will be populated as the overall SCE implementation progresses. As noted above, a clear feedback mechanism should be defined that will allow the pilot team to take information gained during the prototyping, and make relevant updates to the workflows and resulting SOPs and working practices. At the end of the SCE implementation effort, well defined SOPs and Working Practices will be available for the production implementation.

This concludes the SCE Pilot section of the project – once this is complete, the team will have the following documentation available for the priority workflows:

- Workflow Diagrams
- Use Cases
- SOPs & Working Practices

Rather than trying to define “everything” upfront, the methodology allows you to quickly define priority business processes and related documentation. This will greatly facilitate system qualification and user acceptance testing, as well as providing an excellent basis for production user rollout – it will be clear that the SCE has been considered as part of the larger business processes and the user community will surely appreciate that. This will also help to maximize user acceptance, as there is often resistance to the introduction of new processes and systems.

## **SYSTEM VALIDATION**

As a result of the SCE Pilot phase, the team will have a solid understanding of the SCE functionality and how the system will be used within the organization. The formal system validation activities will vary depending on the SCE itself – whether it is a commercially developed product that may be hosted by the vendor, or a simpler server system built in house. In general, the following steps should always be part of the system validation:

- Installation Qualification (IQ) – Verifies system against System Design Specifications
- Operation Qualification (OQ) – Verifies system against Functional Specifications
- Performance Qualification (PQ) or User Acceptance Testing (UAT) – Verifies system against User Requirements

The SCE configuration determined during the pilot will be applied to the QA environment, where IQ, OQ and PQ are performed. Once these validation steps have been completed and any issues addressed, the IQ and OQ steps are performed against the production environment. Once the final configuration settings have been applied and confirmed, the system may be ready for rollout to the users.

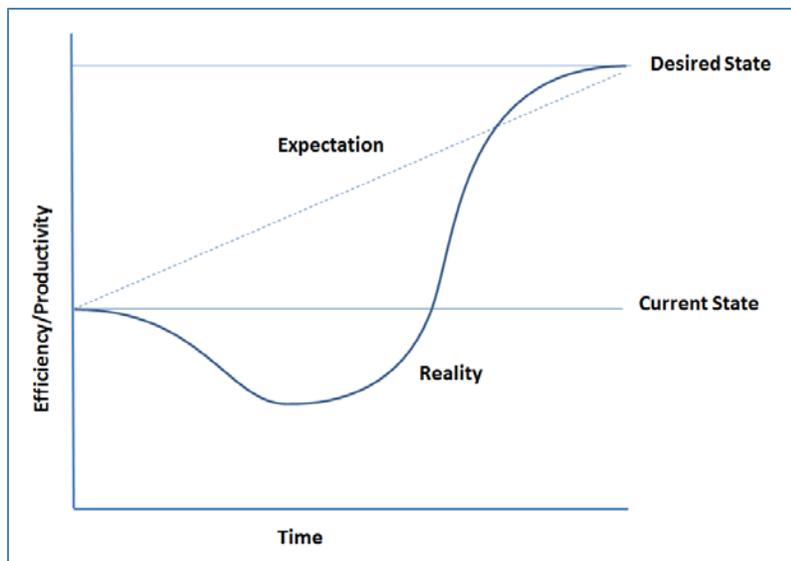
## ADDITIONAL CONSIDERATIONS

With completion of the SCE Pilot and system validation, the team is poised to support the production rollout, but there are a few additional key activities that must be considered to ensure a truly successful SCE implementation.

- Change Management – This should be initiated during the pilot to best prepare the organization for the incoming process changes
- Training – User training on both new processes and new technology – a key part to ensure maximum user acceptance and productivity
- Study Migration – completed or ongoing studies that will need to reside in the new SCE

## CHANGE MANAGEMENT

Implementation of an SCE system often involves a significant change in the way that people approach their work. While the scope of the tasks and the form of the deliverables remains largely similar to their pre-SCE incarnations, the manner in which they are performed and produced will be potentially very different. Given this, continuous change management becomes an essential accompaniment to a successful SCE implementation. To this end, the team must work to institute a change management methodology that will minimize the risk of system rejection by the business user community upon production rollout.



**Figure 3. Productivity Graph**

As illustrated in Figure 3, a productivity dip is expected when new technology and processes are introduced – no technology perfectly addresses all facets of such a complex process. Realistically, the dip in productivity cannot be eliminated, but may be minimized with effective change management.

A formal Change Management strategy should be drafted, with the objective to inform, involve, and prepare all audience groups that will be directly or indirectly impacted by the project. The strategy should identify these different user communities and tailor content appropriately – providing more detailed information to the future heavy system user vs. higher level information to users that will have their work habits minimally affected by introduction of the SCE.

The Change Management strategy may leverage multiple modes of communication such as:

- Corporate Intranet
- Direct email
- FAQs
- Lunch & Learn sessions
- Open “drop in” sessions with SMEs
- Formal training sessions

Note that many of these methods allow (and should encourage) two way communication – it's not only important to communicate the SCE user community about what's coming, but it's equally important for them to know that any concerns they may have are being addressed. Large system implementations can trigger anxiety within the user community – allowing these early and open communications can help alleviate the anxieties and ensure a consistent message throughout the organization, minimizing and impact from the “rumor mill”.

## **TRAINING & SUPPORT**

As described in the change management section, the SCE signifies a major shift in the way the business conducts its day-to-day operations. As a result, training on the system needs to be thorough and conducted in such a way that the knowledge presented in the training session is retained, and found to be of use at the time a user begins to use the production SCE. While SCE training may be available from the vendor, training on any new business processes must be provided as well. The ideal situation would be to have training developed that incorporates the specific business processes with training on the SCE technology – this way, users will see how the two interact, rather than trying to learn the technology and then how the business process is “mapped” to it. This integrated approach will not only save time in training the users, but will help them become more proficient in the system that much sooner, facilitating the production system rollout.

In conjunction with the more formal training, adequate support that is local to the user community will again aid in maximizing user acceptance and speed adoption of the SCE. SCE SMEs would ideally come from the SCE Pilot and prototyping teams, as they will have both an understanding of the “big picture” SCE, as well as more detailed user experiences. These staff would be members of the core user community, but would devote a percentage of their regular “day job” to SCE training and support. Initially, this would be a significant time commitment, but as more users become more experienced with the SCE, time requirements would diminish. The SMEs would be user community first point of contact, so they would also be in a position to collect SCE system and process feedback from the users – as we've said before; these processes will not perfect, and will need to be adapted over time.

## **STUDY MIGRATION**

Prior to production SCE rollout, teams will be working to identify and prioritize studies that will need to have data, programs and related files migrated to the SCE. Initially, a number of priority studies may be identified for initial focus, with lower priority studies being migrated at a later date. Along with designated SMEs, the SOPs and working practices developed during the pilot will be used to guide users through the migration process.

Again, a plan is recommended for this activity – studies identified, migration tasks defined (this may be different for different studies), and timelines set. An overall methodology should be developed and applied for all target studies. For each study to be migrated, the following information should be identified before any work is done:

- Current study file location(s)
- Target study location in SCE
- Types of files to migrate and method to be used
- Any files NOT to be migrated
- Any special handling for this study

## CONCLUSION

Implementing a Statistical Computing Environment (SCE) can bring huge benefits to any life sciences organization, but only if it is recognized that the SCE is as much a **process** change as it is a **technology** change. If the implementation is undertaken with these concepts in mind, the benefits will be felt much sooner than if the organization simply tried to implement new technology with existing processes.

While SCE implementation can seem like a daunting task, using this SCE Pilot methodology, “quick wins” can be had early on in the process. The rapid prototyping can demonstrate how key tasks will be performed in the system, and related documentation – Workflow diagrams, use cases, SOPs, and Work Practices are developed along the way. Teams shouldn’t get stuck on trying to be “perfect” in designing workflows – the idea is that these iterations are performed quickly and thoroughly – but remember that there WILL be multiple iterations and adjustments will be made. Lastly, ensure that Change Management tasks are not ignored – the user community must be ready to accept and work in the new system if the SCE benefits are to be fully realized.

## ACKNOWLEDGMENTS

I would like to acknowledge Chris Decker for his vision for this methodology, Ian Fleming for his contributions to detailing the pilot phase, and Tim Thompson for his expertise in the change management area.

## CONTACT INFORMATION

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

Gary Cozzolino  
d-Wise Technologies, Inc.  
1500 Perimeter Park Dr. - Suite 150  
Morrisville, NC 27560  
gcozzolino@d-wise.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

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