

Paper 237-27

Sounds Like a Good Idea, But What's the ROI? Justifying Your Project and Getting It Approved

John E. Bentley, Charlotte, NC

Abstract:

Too many good ideas for using SAS® Software remain just that: good ideas that are never implemented. Why? Getting an idea approved and funded as a project largely depends on showing that it has real value, and most organizations select projects based on ROI—Return on Investment. ROI is the net profit that a project generates in the form of either additional revenue or reduced expenses. This paper will introduce SAS users to some financial concepts related ROI and suggest how to write a Project Justification Document. The goal is to provide some insight and suggestions on how to get a project approved.

Disclaimer: The views and opinions expressed here are those of the author and not his employer.

The Gravy Train is Over

The days of easy money for IT projects are over, if you were lucky enough to enjoy them. The business cycle has turned down and the bottom line focus has returned. In the corporate strategic plan, controlling costs and increasing revenue have moved to the top of the list of initiatives.

Savvy business leaders, though, realize that the continued investment in strategic technology can be the single biggest differentiator between a thriving business and one that's just treading water or even going under. Smart CEOs are using IT to help position their companies now for the next upswing in the business cycle. Even so, it's no surprise that as part of their cost-cutting efforts organizations are examining IT projects more closely and trying to separate those with the greatest potential for improving the bottom line from those with little or no potential.

Surveys of corporate IT executives continue to document a focus on deriving real business value from IT spending. In the recent past, the value derived from e-business technology, for example, was secondary to having an e-business initiative. Projects with Internet, web, OLAP, or BI in their names were almost guaranteed funding. Now, the approach to IT projects is better aligned with the business and is seen as a tool for implementing the business plan.

Project Justification and ROI

Here's a hypothetical chance conversation in an elevator between a Division CIO and a SAS application developer:

CIO: Bret, I've just come from a meeting where your manager said that you have an idea about making our inventory reports available over the corporate Intranet. It sounds like it will save time and money.

Bret: Yes, it's a great idea... we'd be using cutting edge technology and I know everyone on my team would love to do it.

CIO: Cutting-edge technology? Hmm. Well, it still sounds like a good idea. I've asked that you send in a project justification that includes an ROI estimate and payback analysis.

Bret: Huh?

In recent years, SAS programmers and developers have largely been insulated from concepts related to profitability. For many IT professionals who have fewer than five or six years' experience, the renewed focus on business value

might be a new theme. For the more experienced, it may be only a vague memory. For everyone, well, get used to it. IT projects will continue to get close scrutiny to see if they contribute to profitability. "Profitability" may be calculated differently in different organizations, but it always includes having a positive Return on Investment (ROI) and the sooner the better.

An ROI estimate is a measure of the financial benefits of a project that shows by how much—usually as a percent—new revenues and/or operational cost reductions exceed the investment cost. ROI is derived from a comparison of project costs to project benefits and shows the profit that can be attributed to your project after it is implemented and after the project costs have been deducted. Usually a technique called payback analysis is also incorporated; in this, the length of time it takes to recover the initial cost of a project is factored in. Because of the pace of technological change, managers are looking for shorter and shorter payback periods.

There are a number of financial measures for estimating IT project profitability, and spreadsheet packages will do all the calculations. By applying the same set of techniques across projects, management can accurately compare projects and select the ones with the highest value-added.

A criticism of all four techniques is that while they compare benefits to costs they do so by focusing on current expenses and minimizing maintenance costs. Also, they do nothing to insure that the technology selected and related expenditures are appropriate for the project.

Table 1. Measuring project profitability

Technique	Description
Net Present Value	The impact of time on the value of money. The present value of the expected future cash flow generated by the project minus the cost. Useful for estimating the "real" return from a project.
Internal rate of return	Compares the project's discounted net benefits to the investment yield of a "safe" investment. Often an organization will set a high IRR such as 15 percent so that only the highest value-added projects are funded.
Payback analysis	The length of time it takes to recover the initial cost of a project, usually based on NPV estimates.
Return on investment	The result in percent of comparing total costs to total benefits over a period of time—to be profitable, net new revenues or operational cost reductions from the project must exceed the cost of the investment.
Economic value added	Based on the cost of the money used for the project, by borrowing or other means of capital acquisition. The analysis includes the cost of capital used.

There are three generally accepted models that utilize the techniques presented in Table 1. These models help organizations assess and prioritize technology investments.

Table 2. Models Guiding IT Project Selection

Model	Description
Enterprise ROI	First identify enterprise-level financial targets for new revenue or cost reduction. Then allocate those targets to broad initiatives, which set targets for specific projects. The success of any given project, however, is often dependent on a concurrent project.
Productivity	The project is justified on the basis of employees "doing more with less." Uses goals such as "managers will use 10% less time searching for information." Model results in difficulty in auditing and measuring actual benefits. Often difficult to accurately attribute financial gains to a specific project.
IT Value/Utilization	Focuses on individual project economics and maximizing value of IT investments by managing execution and implementation. Premise is that the technology aligns with the existing IT strategy and has low-cost maintenance. A very common model.

To improve profits, your project must either (a) generate higher revenues or (b) reduce costs, but preferably both. The project can be something completely new, like delivering customer profiles on-demand to a PDA, or an enhancement to an existing system, such as automating the validation process for an existing production report. As long as it adds real measurable value, it's a candidate for approval.

ROI elements

Many items can be used in an ROI estimate but both costs and benefit estimates are needed. You need to take a snapshot of how you're doing business now and then compare that to an estimate of how you'll be doing business when your project is in production. It's nearly impossible to present an ROI estimate without using a baseline to estimate the cost-benefit differential.

One important point to remember: *Costs are incurred up-front during project design, development, implementation, and maintenance—all through the life cycle. Benefits don't accrue until the project goes into production.*

Costs

The first two parts of a four part series by William Roetzheim in *Software Development Magazine* provide valuable information on estimating project costs. Costs can be grouped into three broad areas.

- Time
- People and skill sets
- Hardware and software

Time is obvious. How many people do you need and how many person hours will it take? Multiply people by their hourly rate (or an average). Don't use a best-case scenario, but don't use a worst-case scenario either. Take a hard look at the project and make a good, solid, realistic estimate.

People and skill sets. Do you have the skills the project needs? If not, you'll have to acquire them by either training existing staff or hiring contractors. If you have to train or bring in outside help there will initially be some non-productive time, so be sure to adjust your time estimate to allow for people coming up-to-speed.

Your people-related costs should be adjusted to account for certain "environment factors" that are beyond the scope of this paper but are detailed in the Roetzheim series. Keep in mind that you need the people and skills not only to develop and implement the project but also to maintain and upgrade it over its life cycle.

Hardware and software costs are also obvious. Either you have what you need or you don't. But make sure you know what you need and what you have before submitting your ROI estimate. Finding out a month into the project that you need \$10,000 for DASD will blow away your credibility and may sink the project. Even if you don't need extra DASD this year, what about next year? Forecast your needs three years out.

Benefits

Benefits should be calculated after the costs. It might quickly become clear that the costs are prohibitive, so there would be no useful reason to look at the benefits. There are four categories of benefits derived from IT:

- Time Savings
- Personnel Savings
- Operational Savings
- Revenue enhancement

As a simplified example, let's say that we have an idea to automate a set of ten weekly reports that require fifteen person hours to generate, validate, copy, and distribute through interoffice mail to thirty-five people. Because the process has been stable for a number of months, we want to automate everything: the report generation and validation processes, change the output from listings to spreadsheets, and then email the spreadsheet to the recipients. If there are no validation problems (and there haven't been), the entire process will happen Sunday afternoon and the fifteen person hours are reduced to zero.

Time savings are the amount of time saved by the user community. It is a sort of "time to market" concept and can be difficult to quantify. In our example, the users are the people receiving the reports. We will have to talk with them to get a feel for the value (if any) that they will derive from having the reports in spreadsheet format at 8 am Monday instead of Tuesday when the interoffice mail is delivered.

Personnel savings include improvements in staff utilization such as allowing a staff member to redeployed from one area to another, reducing the number of temporary hires, or slowing the rate of new hiring. Deferred training costs should be included here. Average annual salaries can be used in this estimate. In our example, the person who modifies and submits the program each week and the person who validates the results can be reassigned to other work. Here we save 15 person hours, so we multiply 15 hours by the average hourly salary. But we should reduce that number by an efficiency factor of 30% because not all time saved will be redirected productively.

Operational savings can be unexpectedly high as you "speed up" a system to run a job faster or "scale up" to handle more work in the same amount of time. To estimate savings in this area, the Systems Administrator can provide an estimate of how much it costs to run the system on an hourly basis. In our example, the project may require a

substantial rewrite of the existing report generation code and that may allow us to make the code more efficient. Time saved here, though, will have to be subtracted from the time it takes the new validation routine to run.

Operational savings also include deferred software and hardware costs. It may be that one of our motivating factors for our project is that we heard a rumor that the IT group was considering purchasing an OLAP software package to do what our project does. Since we already have SAS Software, we don't have to purchase the new package, install and implement it, or train users. We also save the paper that the hardcopies are printed on. Finally, if the reports deal with inventory, there may be savings attached to being able to place a restock order Monday morning instead of Tuesday afternoon.

Revenue enhancement from an information analysis or delivery project can be difficult measure. If a manager uses your information along with other information (and his own experience) to make a decision that generates \$250,000 more revenue, it's difficult to assign a percentage to any one piece of information that he used. But if your project will result in faster turn-around or higher response for a marketing campaign, it's easy to assign a dollar value to that "lift".

A Hypothetical Case Study

Transnational Cooperation is an international economic development organization. Its current reporting environment is adequate but requires an average of 500 person hours per month to support a user community of 800 individuals. A proposed SAS-based, web-enabled system—Project KnoAll—will dramatically reduce support costs and improve the timely availability of information.

What follows are cost-based financial indicators that can be used to decide whether or not to approve with the project. Benefits (savings) from increased productivity and better decision-making are not included.

Assumptions:

1. The project will take one FTE and one contractor 1000 hours to develop and implement.
2. Contractor labor costs are \$80 per hour; fully weighted FTE labor costs are \$62 per hour.
3. In production, the solution will require support of 80 person hours per month or ½ FTE.
4. Labor costs will increase 4 percent annually.
5. Hardware is a one-time expense.
6. Software license renewal is 40 percent of the first year license and will increase 4 percent per year.
7. The solution's life span is three years.
8. A "safe" investment yields 7% (the discount rate).
9. Costs will be covered by internal cash flow and no money will be borrowed; there is no cost of funds.
10. The existing reporting system will be turned off after the new system has been in production for three months.

Table 3. Project KnoAll Costs

	Year 0	Year 1	Year 2	Year 3
Hardware	\$ 250,000	\$ 0	\$ 0	\$ 0
Software	159,000	64,000	67,000	70,000
Labor	142,000	62,000	65,000	68,000
Total	\$551,000	\$126,000	\$132,000	\$138,000

Table 4. Cost Comparison

	Year 0	Year 1	Year 2	Year 3
Project KnoAll	\$551,000	\$126,000	\$132,000	\$138,000
Current System	432,000	449,000	464,000	478,000
Net Savings	None	\$323,000	\$332,000	\$340,000

A comparison of the Year 0 initial investment in the new solution to the current solution is not needed. The annual cost of the current reporting system is a budgeted expense. But because some of its expenses may be avoided by shutting down the system before the end of the fiscal year, it would be useful to include these operational savings in the benefits estimates.

Table 5. Summary of financial measures

	Year 1	Year 2	Year 3	Total
Net Savings	\$323,000	\$332,000	\$340,000	\$979,000
Discounted net at 7 %	301,869	289,982	277,541	\$860,657
Internal Rate of Return				36 %
Payback Period				1.9 years
Return on Investment				156 %

The net savings produced by the project is \$979,000 but in today's dollars the net present value (NPV) of the savings is actually \$860,657, which reflects an annual depreciation of 7 percent in value of money over the project life span.

$$NPV = (323,000/1.07) + (332,000/1.07^2) + (340,000/1.07^3)$$

The internal rate of return generated by the project is 36 percent, well above the 7 percent yield used in the NPV calculation. If the IRR was 7 percent or less, then there would be no real financial reason for the investment—the same return could have been obtaining by a less risky investment.

$$IRR = (323,000/1.36) + (332,000/1.36^2) + (340,000/1.36^3)$$

The payback period is 1.9 years, or about 1 year and 11 months. This means that the discounted net savings will offset the initial investment faster than the new system's life span. If the payback period were three years or more, then the new system would need to be replaced before the investment generated a return.

$$\text{Payback Period} = 551,000 / (860,657/3)$$

The new system generates a return on investment of 156 percent, meaning that the financial benefits of the new system exceed the investment by 56 percent. If the ROI were 100 percent or less, then the benefits would be equal to or less than the cost of the project.

$$ROI = 860,000 / 551,000$$

In summary, based on cost considerations alone the project should be implemented:

- The IRR of 36% is much greater than a safe investment yield rate of 7 percent.
- The payback period is 1.9 years, over a year faster than the anticipated system life span.
- The ROI is 156 percent.

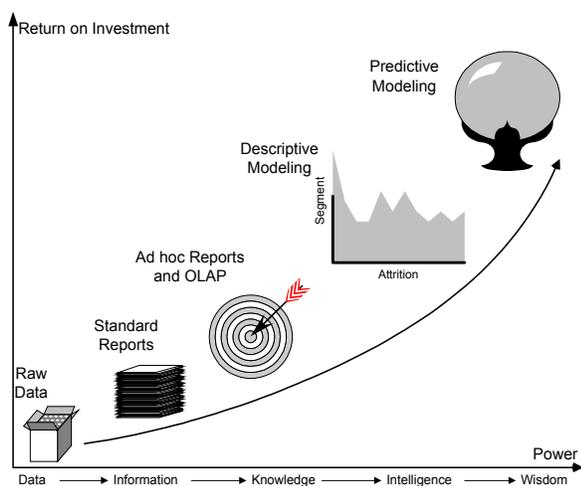
Which Projects Will Be Approved?

The surest way to getting a project approved is to learn your organizations priorities and strategies and then propose a project that supports them. A project that delivers well-defined business results that support strategic goals should always make the first cut in a budget review. Finding a partner on the business side to help you develop, focus, and refine your initial concept will also improve your chances. Projects that are jointly developed by business and IT staff are always more likely to be approved and projects that already have business support get priority.

Projects related to data and information differ in the business value they deliver. Capturing raw data is good, but in and of itself doesn't add value; analyzing and reporting adds value to data. Discovering patterns, running complex statistical analyses, and model building all add even more value.

As a joint SAS Institute/EMC White Paper describes it, there is a continuum along which data moves to first become information, then knowledge, then intelligence, and finally wisdom. It's at the higher levels that companies gain the highest ROI in information gathering, processing, reporting, and analysis. Figure 1 depicts how moving from having raw data to performing predictive analysis increases the power of information and, consequently, ROI.

Figure 1. Information Power = Higher ROI



Information's value also increases as more people have access to it, and delivering the right information at the right time to the right people is **extremely** valuable. Applications that deliver information are high value-added projects, and the Web provides a perfect channel to expand information access to front-line supervisors and customer-facing staff. Applications that automate processes and tie together data analysis and delivery are also high value-added.

The Project Justification Document

A Project Justification Document is not a formal IT Proposal. IT Proposals are usually required only for major system implementations or modifications. ("Major", of course, is defined differently by different organizations.) An IT Proposal is much more time and research intensive than a Project Justification and is usually a formal project in itself that includes team members outside the IT group. A justification document, on the other hand, provides management with a concise evaluation of a significant problem, suggests options, presents the preferred solution with cost and benefits, and asks for a go-ahead. A Project

Justification Document, however, may lead to and serve as the basis of an IT Proposal.

The Project Justification Document is exactly what it says: it justifies the project and says on why it is important. Here are some useful definitions:

- Project: An extensive undertaking requiring an extended, focused, concerted effort.
- Justification: A condition that demonstrates a concept or action to be right, valid, or just.
- Document: A written or printed paper that contains information presenting decisive evidence

Based on our definition of terms, we need to identify a basic message, the message contents, and a writing style. The document's language and style should be short, clear, and concise. More is not better. Remember that non-technical business and budget managers may be reading it.

- Be direct and unequivocal—don't say "in first quarter 2002, say "in March 2002", "
- Quantify when appropriate and possible—don't say "several dozen users", say "36 known users".
- Be specific-- don't say "this project is important", say "the outcome of this project will enable <person or group> to achieve their goal of <goal>.

The document should be only two or three pages long. Each section (detailed below) should have a short heading and be no more than two or three paragraphs and a table if appropriate. Where possible, use bullet points instead of a paragraph. Use power words and phrases:

- Effectiveness; the ability to work smarter; do more with less.
- Improved productivity wherein fewer people do the job in a shorter time period; do more with less.
- Ease of use; requiring minimal training; having a steep learning curve.
- Reuse underutilized hardware and software that the organization already owns.
- Scalability; the ability to accommodate growing volumes of data, demands for speed, and increasing numbers of users.
- A neutral architecture that conforms to industry standards and supports multiple platforms.
- Data exchange and integration with other applications.
- Reuse and sharing of the project's components and modules within the development organization and for future projects.

Writing a Project Justification Document

Know Your Audience

The purpose of the justification is to persuade decision-makers that they should approve and fund the proposal by allocating time and money in the budget. But who are these decision-makers? It's entirely possible that your own supervisor or manager is not one of them. Instead, they may be your organizations senior managers, financial group managers, and budget analysts. Very possibly, they don't know or understand the project's technology, so you have to be very clear and specific in your writing.

What do the reviewers and decision makers want to know? They want to know the same things you want to know when you are trying to decide whether or not to purchase a product or service for your own use:

- What specifically is it?
- What specifically will it do?
- How much does it cost?
- How much does it cost to maintain it?
- When will I get it?
- Can I trust the seller?

Write an Outline

Writing an outline is similar to flowcharting: most people agree that it's valuable, but it's usually not done. In this case, an outline helps organize important points, identify which information to include, and avoid things that obscure the message.

The major sections of the document are:

- Alignment with a business objective or a strategic goal. ("Here is our business goal, and this project will help us achieve that goal.")
- Cost estimates. ("This is what it will cost to do it and what it will cost to maintain it.")
- The approach. ("Here's how we'll do it.")
- Key deliverables and a schedule. ("Here is what you will get and when you will get them.")
- A project control plan. ("Here is how you know you are getting the work done on time and within budget.")

Flesh out the Outline

Align with a business objective or a strategic goal. Find out what objectives and goals are hot buttons with the decision-makers and think of ways that your idea will support those. Generally, this means doing things better, cheaper, or faster than they are now. You need to increase revenue or decrease costs. The description of the project here is in broad terms. Start with a "Project Objective Statement":

"To [implement, complete, activate, etc.] [what—a short description] by [month, year], which will enable [business group, function, or process] to [increase, decrease, etc.] [a business-related objective or goal]."

Cost Estimates. The cost estimates here are neither exceedingly detailed nor agonizingly accurate. Although accuracy definitely counts, senior managers and financial analysts are comfortable with estimates and realize that you're projecting costs up to six months or more in advance. Always **give a good faith estimate** and let them know that. Depending on how much you're asking for, you should probably round your estimates to thousands or hundreds of dollars. At a minimum, the cost estimate must include the total amount requested, broken down into personnel expenses and non-personnel expenses for a specific time period, usually the coming fiscal year.

Personnel expenses include the salary cost of employees who will be working on the project only the period of time that the employees will be assigned to the project. That is, if a person is working full-time on the project for three months, only one-quarter the annual salary is requested for the project. Also include cost estimates for contractors who will be needed for the project. (Obviously, by the time you get to this point you will need to know the skill sets that your project will take and the number of man-hours it will require.)

The amount for non-personnel expenses must include **all** other expenses associated with the project. This includes

hardware and software acquisition and licensing, training, travel to the training, reference manuals, the cost of setting up work cubicles, long-distance phone calls, even for late-night dinners. Try to think of **everything**. Don't knowingly underestimate an expense and blind-side your managers later with a request for something you should have foreseen, like \$2,500 to send a developer to Java training when your training budget was \$59.95 for "Learn Java in 15 Days".

It's important that your cost estimates not only include costs to get the project completed, but annual estimates of what it will cost to keep it in production over its life-cycle. Is there an annual license renewal associated with the software, or as the database grows will you have to purchase more disk space? If so, get those costs out in the open now. It helps establish your credibility, and by showing that you've done the research you'll somewhat reduce the likelihood that anyone will question your numbers.

The Approach. Briefly explain how you identified the need for the project and the steps you'll take in the design, development, testing, and implementation phases. You should include a paragraph about how you've worked closely with a business partner in developing this solution. (You did, right? How else do you know that this is the best solution?)

It's important to include a few sentences on any significant risks and how they will be prevented or mitigated. **Be up-front about the risks, but don't dwell on them.** If, for example, you need a UNIX server with a specific configuration but you've found an underused box that fits your specs and your cost estimates assume that you'll get access to it, you must say what the consequences will be if you don't—either you'll be delayed while you find another box or the cost goes up when you buy a box or the project is stopped. If you need a person with a specific skill set like Java and she's working on another project now, what will happen if she isn't available when your project needs her?

A description of how you will assure that the project will deliver results is also necessary. This could include a concise overview of the software development and/or project management methodologies that will be used, the extent of end user involvement in development and testing, project progress reporting, and post-implementation follow-up and evaluation. This section should not be exhaustive treatment of any of the items mentioned. The intent is to help decision-makers understand that you have thought through the project and understand what needs to be done to deliver what you are promising.

Key Deliverables and a Delivery Schedule. This critical—it's where you spell-out what they are getting for their money and when they will get it. Deliverables are **tangible** products that the project will generate; there are intermediate deliverables (e.g., a beta version of a software) and final deliverables (e.g., fully tested and documented software placed in production). Always include the final deliverable and delivery date. **Don't over promise!** A delivery date that appears "overly aggressive" will make people question your credibility and ask more questions.

Depending on the project, intermediate deliverables could include:

- Business and System requirements documents
- Hardware acquisition specifications and installation
- Software installation
- Database design and implementation
- System and user acceptance testing
- User documentation
- User training

The Project Control Plan. Briefly explain here how you will keep the project activities on track and insure that it will be finished on schedule. This section should include a few sentences each covering:

- Project plans and schedules. (Avoid details.)
- A short table of key milestones and dates.
- Key performance indicators for monitoring and tracking progress.
- The process for taking corrective actions when needed.
- What periodic meetings you will have, who will attend them, and when, how, and to whom you will communicate project status.

Double check the numbers and get a peer review

Budget numbers are closely checked, so don't let a simple mistake in arithmetic undermine your proposal. Even though many of the people reviewing your proposal may not be IT professionals, they'll know enough to know when they're being sandbagged. Don't use clearly unrealistic estimates because credibility is the very first criteria a proposal must pass.

Get a peer review from someone who isn't familiar with the business issue you're address or the technology you want to use. After they read it, ask them to tell you what the project is, what it is for, what the business benefits are, and why it's important. If your reviewer can't answer these questions, the people reviewing your proposal won't be able to either.

Summary

An idea needs to be funded before it becomes a project, and funding depends on the anticipated Return on Investment (ROI) and other financial measures such as the Internal Rate of Return and Payback Period. Calculating these measures is not difficult, but it's critical that the Project Justification Document presenting them be clear, concise, and as accurate as possible.

Although the Project Justification Document may be only the first step in the approval process, it's a critical first step. For that reason, it's important to keep in mind that non-IT staff will very likely review your proposal and that the Document has to clearly describe the project, what it will accomplish, and how much it will cost in addition to how much it will add to the bottom line.

References and Resources

Hummingbird. "Enterprise IT Value: Beyond Data Warehousing and ROI."
www.hummingbird.com

Information Discovery, Inc. "Measuring the Dollar Value of Mined Information".
www.datamine.aa.psiweb.com/infoval2.htm

Resources Management Systems, Inc., on-line tutorial. "Get Your IT Project Funded – 5 Steps to Improve the Odds".
www.rms.net/tut_proj.htm

Roetzheim, William H. "Estimating Software Costs." Software Development Magazine, 4-part series, October 2000 to January 2001.
www.sdmagazine.com/articles/2000/0010/0010d/0010d.htm

SAS Institute and EMC, Inc. "SAS and EMC: Working Together to Turn Data into Knowledge." White Paper.

The Beta Group. "IT Budgets in Uncertain Times, Part 1: Managing The Legacy "Entitlement Programs". Working Note, 2001.

Wu, Jonathan. "Calculating ROI For Business Intelligence Projects." Base Consulting Group, December 2000.
www.baseconsulting.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

About the Author

John E. Bentley has used SAS Software for fifteen years in the healthcare, insurance, and banking industries. For the past three years he has been with the Corporate Data Management Group of Wachovia Bank with responsibilities of supporting users of the bank's data warehouse and data marts and managing the development of SAS client-server applications to extract, manipulate, and present information from them. John regularly presents at national, regional, and special interest SAS User Group Conferences and local SAS User Group meetings. He is co-chairing the SUGI27 Data Warehousing and Enterprise Solutions section and was Program Chair for the 2000 and 2001 Data Mining SAS User Group Conferences.

Contact Information

John E. Bentley
Corporate Data Management Group
Wachovia Bank
201 S. College Street
Mailcode NC-1025
Charlotte NC 28288
704-383-2686
John.Bentley2@FirstUnion.Com

