

# Survival Strategies and Solutions for the New Millennium -- Year 2000

**Kirk Paul Lafler, New Millennium Solutions**  
**Sunil Kumar Gupta, New Millennium Solutions**  
**Charles Edwin Shipp, New Millennium Solutions**

## ABSTRACT

As the turn of the century approaches, critical challenges face organizations world-wide. Will mission-critical computer applications be equipped to handle the numerous calculations and comparisons required by businesses to successfully operate? And how can organizations smoothly transition into the next century without impacting current operations and staffing levels?

If estimates are accurate, as much as 90 percent of all business applications will be impacted in some way by the coming millennium. A successful strategy is needed to ensure cost-effective, timely, and lasting solutions.

## INTRODUCTION

Organizations world-wide are confronted with a daunting challenge: how to successfully transition date-dependent computer applications from this century to the next. And as if this challenge wasn't enough for the already overburdened IS organization, a host of other questions appear larger than life. Questions such as how many applications will be affected? Who will perform the conversion tasks? How long will it take? How much money from the budget will be required? What levels of staffing should be placed on the Year 2000 effort? What level of business risk will the organization face? How soon should the effort begin? What is the best method to get started? Most, if not all, organizations will confront these questions and more. And the questions won't stop there, since this monumental task must be completed without impacting current business demands.

This paper presents a comprehensive approach to minimize disruption while reducing risk to an organization. Numerous strategies and solutions using the SAS software will be presented to effectively transition mission-critical applications smoothly into the next century.

## ORIGIN OF THE PROBLEM

The Year 2000 problem can be traced to the way many computers and programs handle dates by using a two-digit year notation method, for example 04/01/97. The

two-digit year notation has come about primarily to save storage space, and perhaps to a lesser degree to reduce the number of keystrokes incurred by data entry personnel. The two-digit year affects the way data is manipulated, particularly when performing subtractions and comparisons.

For example, the year "1980" is stored as "80" in most data files, and "2000" will be stored as "00". If we were to calculate how many years between 1997 and 1980, the computer would correctly subtract 80 from 97 deriving 17. But what happens when performing the same calculation in the year 2000? The computer would subtract 80 from 00 resulting in an incorrect answer of -80. Errors of this type could affect any calculation that derives or uses time spans, such as when performing simple interest calculations.

The problem doesn't end there. Many online applications allow only for a two-digit year where the value "2000" would be entered as "00". Other areas where problems could occur involve printing hard-copy reports. And add to the mix those application programs that process dates beyond the current date, (i.e., budgeting systems). The areas vulnerable to the problems associated with the Year 2000 are extensive.

## THE GLOBAL IMPACT

On September 6, 1993, ComputerWorld published the now famous Doomsday article by Peter de Jager. He had this to say about the global impact, "*The problem is twofold: the date issue itself and, more importantly, our reluctance to address the problem.*" It is a world-wide problem. The problem can occur in large or small companies, domestic or international businesses, and in any industry. These global concerns are only accelerated with the mushrooming commercial growth of the Internet.

## ASSESSING THE PROBLEM -- IMMOVABLE DEADLINE

The first thing to realize is that simply ignoring the problem won't make it go away. The problem must

start to be addressed today or there won't be enough time

to solve it. *"Don't get bogged down with the methodology -- just get started,"* says Janice VandenBrink, VP of settlement and reporting systems at Visa International.

Should the conversion of mission-critical application programs not be met with "sound" practical solutions, potential catastrophic events could, and most probably will, result. The severity of the problem is heightened since date fields, for lack of a standard, are used in a variety of ways to perform such common processes as payroll, profit/loss statements, invoices, loans, interest calculations, security access systems, telephone switching, and numerous other equally important tasks. There are some estimates that as much as 90 percent of all business programs will be impacted in one way or another by the pending century change.

What makes matters worse is that many organizations are unaware of the magnitude of the problem. Too often it is treated as a maintenance issue and given little or no priority and budget to adequately address the issue. To minimize the importance of these activities can place the entire organization's future at risk.

So how should an organization go about assessing the problem? It can begin by first having a firm grasp of the organization's purpose and values. Once this is articulated, approach each strategic business unit (functional area such as accounting, payroll, inventory, etc.) with an understanding of how much resources will be devoted to each. With this knowledge, assess the capabilities and limits of current staffing levels. Only when this is performed can an honest assessment be given of the magnitude of the conversion.

### **LOOKING AT THE POSITIVE**

As with most projects of this scope it is important to look for a silver lining, especially when involvement requires a sizable budget outlay. As International Business Machines said in a 1995 Executive Report, an expenditure of this magnitude can be explained to stockholders and management simply as, *"We still have a viable business, one that accepts billing, correctly computes interest and invoices, and processes payroll in a timely manner. Not all of the competition will be able to make that claim."* Companies may even look upon this as an opportunity to upgrade their entire system and software applications.

### **FREQUENTLY ASKED QUESTIONS (FAQs)**

### **DERIVING THE SOLUTION**

Surviving the conversion of programs and data to handle the new millennium begins by deriving a "workable" tailored solution that fits a specific organization's business climate and culture. Nothing less than a comprehensive solution will do. Consequently, prior to coding and/or converting a single field in a record, the following approach should be adhered to.

First, have a firm understanding of the extent of the problem. This means that every aspect of the organization's business should be examined. What type of processes occur? How is data captured? Why is the data needed? How is the data used? Who uses the data? What application programs use these data sources? Only when these questions are answered can an organization be certain of the importance of an application and its data to the organization.

Next, prepare a detailed conversion plan with strategies to conduct a successful conversion. The *strategic* conversion plan includes such items as a comprehensive list of affected application programs, program sources, list of date fields, data sources, and program and application dependencies.

Finally, test and implement the plan to ensure a comprehensive and lasting solution. The conversion plan is used to design, code, and test application program conversions. Conversion efforts can be performed in-house, outsourced, or a combination of both.

### **THE COST/BENEFIT**

Some estimate that the costs associated with fixing the Year 2000 problem could total hundreds of billions of dollars. Ken Orr, principal at the Ken Orr Institute, and Larry Martin, president of Data Dimensions, Inc. estimate that Fortune 50 organizations will have to spend about 35 to 40 cents per line of code to convert all their existing systems to accept the change from the year 1999 to 2000. Peter de Jager states in his 1993 Doomsday article, *"that translates into about \$50 million to \$100 million for each company."*

The benefit of performing the transition to the Year 2000 may result in a greater knowledge of software and program inventories. It may be possible to even reduce and/or eliminate duplicate, as well as nonfunctional, code from swelling inventories. In any case, the process will allow a closer look at what types of data are being collected and how this data is functionally handled within the organization.

A wise person once said, *"there is no good time to have a problem with your computer or the software you are*

*using."* Since the Year 2000 and its associated problems will affect tens of millions, and most probably hundreds of millions, this statement seems most appropriate. As is often the case with complex and/or demanding problems, an array of questions arise. The following Frequently Asked Questions (FAQs) address a few of the more interesting questions and attempt to shed some light for the reader. The reader is advised to look elsewhere (i.e., Internet) for a more detailed and comprehensive list of FAQs.

**Q1.** What will happen if the Year 2000 issue isn't corrected?

**A1.** Any computer calculation or comparison involving a date could yield an incorrect answer.

**Q2.** Is the problem with Year 2000 a hardware or software problem?

**A2.** It is primarily a software problem.

**Q3.** Is there anything that will need to be done to my hardware?

**A3.** Computer users should read their users manuals and/or contact their service representative or sales vendor to determine whether their computer hardware can handle the change of century.

**Q4.** Does this problem only occur on mainframe systems and/or legacy applications?

**A4.** No. Any system or programs can be affected if two-digit year notation is used.

**Q5.** Is 2000 a leap year?

**A5.** Yes. To help you determine whether a given year is a leap year, apply these simple rules.

① When the year is evenly divisible by 4 it is a leap year, except for years ending in 00.

② A year ending in 00 is a leap year if it is evenly divisible by 400.

**Q6.** How can one proceed in solving the various Year 2000 problems?

**A6.** Apply the following three step rule for every Year 2000 problem:

① Prepare by taking inventory of all application programs.

② Implement by identifying, correcting code and data, and verifying.

③ Deploy by performing system tests and moving testing systems into production.

**Q7.** Will SAS software handle dates in the twenty-first century?

**A7.** Yes. SAS software is already equipped to handle the date transition. If dates in your file consist of a four-digit year, then SAS can accept and display these with little or no difficulty. If dates in your file consist of a two-digit year, then the use of the

YEARCUTOFF= system option may be necessary to differentiate the desired century.

## CONCLUSION

The following recommendations can be made on how an organization could approach Year 2000 planning. The most important thing to remember is not to procrastinate. Start by creating a schedule of activities to be performed and be realistic as to the level of resources available.

*Table 1* is not meant to be all-inclusive, but to be used as a guide to how an organization could go about planning their activities.

As the turn of the century approaches ever so close, will mission-critical computer systems be ready to handle the numerous calculations and comparisons required to successfully operate into the next century? Organizations that procrastinate to become Year 2000 compliant, risk not only losing their clients to competitors, but may risk survival themselves.

The time to begin is now, so just get started.

## ACKNOWLEDGEMENTS

The authors would like to thank Janet Bosomworth, SUGI Conference Chair, and Chris Toppe, Data Warehousing Section Chair for their support and encouragement during the development of this paper.

## REFERENCES

Adhikari, Richard, "Approaching 2000." Information Week, October 7, 1996, pps. 36-48.

de Jager, Peter, "Doomsday," ComputerWorld, September 6, 1993.

Elam, Amber H., "SAS Dates: Everything You Always Wanted to Know, Including the Year 2000," SAS Observations, Third Quarter 1996, pps. 23-30.

SAS is the registered trademark of SAS Institute Inc. in the USA and other countries. Other brand and product names are registered trademarks or trademarks of their respective companies.

#### **AUTHOR BIOGRAPHY**

Kirk Paul Lafler is senior consultant and co-founder of New Millennium Solutions. Kirk is a SAS Institute Quality Partner with nearly twenty years of SAS programming and training experience. His areas of expertise include information-based system analysis, design, and development, computer training, and programming using base-SAS, SAS/SQL, SAS/FSP, SAS/AF, Screen Control Language, FRAME, and SAS/EIS software.

Sunil Kumar Gupta is senior consultant and co-founder of New Millennium Solutions. Sunil is a SAS Institute Quality Partner with nearly seven years of SAS programming experience. His areas of expertise include programming using base-SAS, SAS/STAT, SAS/FSP, SAS/AF, Screen Control Language, and SAS/GRAPH.

Charles Edwin Shipp is senior consultant and co-founder of New Millennium Solutions. Charles is a SAS Institute Quality Partner with nearly fifteen years of SAS programming and training experience. His areas of expertise include base-SAS, SAS/AF, Screen Control Language, SAS/OR, SAS/ETS, statistical visualization, and JMP applications and training.

The authors welcome comments and suggestions. They can be reached at:

**New Millennium Solutions  
P.O. Box 1390  
Spring Valley, California 91979-1390  
(619) 670-7638**

<b>Year 2000 Organization Plan</b>		
<b>Task</b>	<b>Status</b>	<b>Estimated Completion Date</b>
Obtain Senior Level Management Sponsorship of Project		June 1997
Assign Year 2000 Project Manager		June 1997
Develop Project Planning Tool		July 1997
Create System Inventory and Prioritize		August 1997
Determine Resources and Costs		September 1997
Develop Detailed Project Plan		November 1997
Develop Alternative (Contingency) Plan		November 1997
Select Vendors (For Outsourcing)		As Needed
Begin Modification of Test Programs/Systems		June 1998
Conduct Unit Testing		July 1998
Evaluate Unit Test Results		August 1998
Conduct Integration Testing		October 1998
Evaluate Integration Test Results		November 1998
Perform Data Conversion		January 1999
Final Testing of Programs/Systems		March 1999
Train Users Accordingly		April 1999
Implementation of Systems into Production		June 1999
Follow-up/Make Corrections		March 2000

**Table 1. Sample Year 2000 Organization Plan**