

University of Mississippi

eGrove

Guides, Handbooks and Manuals

American Institute of Certified Public
Accountants (AICPA) Historical Collection

1998

Solving the year 2000 dilemma

Sandi Smith

Follow this and additional works at: https://egrove.olemiss.edu/aicpa_guides



Part of the [Accounting Commons](#), and the [Taxation Commons](#)

AICPA Top Technology Series



Solving the Year 2000 Dilemma

SECOND EDITION

Sandi Smith, CPA, CMA, CDP

AMERICAN INSTITUTE OF CERTIFIED PUBLIC ACCOUNTANTS

AICPA

Solving the Year 2000 Dilemma

SECOND EDITION

AICPA

NOTICE TO READERS

This book is provided for educational and informational purposes only and does not provide legal, accounting, or other professional advice or services. Use of any information in this book does not create a professional relationship. You should not act on the information provided in this book without seeking additional legal, accounting, or other professional counsel. The authors and publishers do not warrant that by following the advice herein the reader's systems, products, or services will be year 2000 compliant.

Solving the Year 2000 Dilemma does not represent an official position of the American Institute of Certified Public Accountants.

AICPA Top Technology Series



Solving the Year 2000 Dilemma

SECOND EDITION

Sandi Smith, CPA, CMA, CDP

AMERICAN INSTITUTE OF CERTIFIED PUBLIC ACCOUNTANTS

AICPA

Copyright ©1998 by
American Institute of Certified Public Accountants, Inc.,
New York, NY 10036-8775

All rights reserved. For information about the procedure for requesting permission to make copies of any part of this work, please call the AICPA Copyright Permissions Hotline at 201-938-3245. A Permissions Request Form for emailing requests is available at www.aicpa.org by clicking on the copyright notice on any page. Otherwise, requests should be written and mailed to the Permissions Department, AICPA, Harborside Financial Center, 201 Plaza Three, Jersey City, NJ 07311-3881.

1 2 3 4 5 6 7 8 9 0 PP 9 9 8

Library of Congress Cataloging-in-Publication Data

Smith, Sandi.

Solving the year 2000 dilemma / Sandi Smith. — 2nd ed.

p. cm. — (AICPA technology series)

Includes bibliographical references.

ISBN 0-87051-241-2

1. Software maintenance. 2. Year 2000 date conversion (Computer systems) I. Title. II. Series.

QA76.76.S64S62 1998

005. 1'6—dc21

98-40609

CIP

DEDICATION

To my dear Mom and Dad, the best parents on the planet.

ACKNOWLEDGMENTS

I am lucky to have such a great editor at the AICPA on not only the revision of this y2k book, but also the top 10 technologies books. A huge thanks goes to Murray Schwartzberg for his part in producing this book and seamlessly orchestrating the rest of the AICPA team: Nancy Cohen, technical editor, who kept me honest and accurate; Mary Mooney, copy editor, who corrected my grammar among other things; Peggie Burns who keyboarded all those corrections; Thomas Ray and Gretchen Fischbach, standards editors, who checked to see that we hadn't contradicted an accounting principle; and Jeanmarie Brusati, production supervisor, who made the book look good in print. A big thank you goes to all of these hard-working team members. One more special thanks goes to Marie Bareille for her efforts in making this information internationally available.

TABLE OF CONTENTS

SECTION I	The Year 2000 Dilemma	1
CHAPTER 1	It's Bigger Than a Breadbox	3
CHAPTER 2	Don't Take It Personally, But	9
CHAPTER 3	Strictly Business	19
CHAPTER 4	Inside and Outside the Organization	43
CHAPTER 5	2000 Is a Leap Year	45
CHAPTER 6	And Now for the Silver Bullet.	47
SECTION II	Year 2000 Issues	49
CHAPTER 7	Awareness and the Antics That Follow	51
CHAPTER 8	Accounting Issues.	57
CHAPTER 9	Legal Issues	61
CHAPTER 10	Life Insurance on Your Business?.	69
CHAPTER 11	Human Resource Issues	71
CHAPTER 12	Project Management Issues.	77
CHAPTER 13	Enterprise Issues	83
SECTION III	Year 2000 Solutions	87
CHAPTER 14	Project Initiation	89
CHAPTER 15	Systems Assessment	95
CHAPTER 16	System Renovation	107
CHAPTER 17	Implementation	115
CHAPTER 18	Date Modification Options	119
CHAPTER 19	Y2k Consultants.	123
CHAPTER 20	Tools	129
CHAPTER 21	Final Tips.	137
CHAPTER 22	Life After the Year 2000	143
APPENDIX A	Systems Inventory Checklist.	145
APPENDIX B	Sample Project Plan	149
ENDNOTES	153
GLOSSARY	165

THE YEAR 2000 DILEMMA



Dilbert reprinted by permission of United Features Syndicate, Inc.

This book is dedicated to educating the accountant about the year 2000 (abbreviated throughout this book as y2k) dilemma. Divided into three sections, the book will take you through:

1. The dilemma
2. Its issues
3. The solutions

I'll discuss the dilemma at length because you'll find that the more you know about the problem, the better you'll understand its reach on our businesses as well as on our personal lives. In Section II, "The Year 2000Y2k Issues," I'll spend time on the issues from various perspectives: accounting, legal, human resources, and enterprise-wide, to name a few. In Section III, "The Year 2000 Solutions," I'll walk you through the steps of a project and look at vendors and tools available on the marketplace.

The six chapters of Section I, "The Year 2000 Dilemma," present the y2k problem and its major wrinkles and problems. Chapter 1, "It's Bigger Than a Breadbox," gives a description and definition of the scope of the problem. Chapter 2, "Don't Take It Personally, But. . .," will take a look at how each of us will be affected personally by the y2k dilemma. In chapter 3, we'll take a business perspective, discussing the impact you might not normally think of on hardware, software, and systems. Chapter 4, "Inside and Outside the Organization," will help us look outside our organizations for y2k problems; chapter 5 "2000 Is a Leap Year," will warn us about the leap year day in February 2000; and chapter 6, "And Now for the Silver Bullet," will caution us about "silver bullets."

CHAPTER 1:

IT'S BIGGER THAN A BREADBOX

The y2k problem that the y2k dilemma creates couldn't possibly be any big deal to fix. It's so simple that a third-grader can understand it. Right?

The y2k dilemma is just a bunch of computer programmers' talk to get attention. Right?

My small business won't be affected. The y2k dilemma is only a "Fortune 500-sized" problem. Right?

The y2k problem is just the next buzzword for consultants so they can get more work. Right?

Regardless of what we do, nothing bad will happen on January 1, 2000. Right?

Wrong. Wrong. Wrong. Wrong. And wrong again. These are just some of the myths floating around that are masking the complexity and enormity of the y2k dilemma.

Two facts are irrefutable:

1. Our society is tremendously dependent on computers. Most people will not realize how true this statement is until they finish reading this book.
2. Most computers were designed and sold to businesses and consumers with a fatal flaw: one that will not let them correctly process data past December 31, 1999.

The costs to fix the problem are staggering. Gartner Group's estimate of \$300 billion to \$600 billion to fix the problem globally is widely quoted. Journalists have compared the estimate to this century's largest natural disasters such as the Kobe earthquake (\$100 billion), the Los Angeles earthquake (\$40 billion), and Hurricane Andrew (\$30 billion).^{1,2} Even if you combine these disasters, the sum does not approach the cost of fixing the y2k problems.

A Merrill Lynch paper goes a step further to put the y2k dilemma "in perspective" with this century's major events (see table 1.1).³

TABLE 1.1

<i>Event</i>	<i>Estimated Cost in Billions of Dollars</i>
World War II	\$4,200
Year 2000	\$600
Vietnam Conflict	\$500
Kobe Earthquake	\$100
Los Angeles	\$60

According to *USA Today*, David Wyss, chief economist of Standard and Poor's DRI, a consulting group, says economic growth could be reduced by 0.3 percent in 1999 and 0.5 percent in 2000 due to the y2k dilemma.⁴ That's about the same expected size of the Asian crisis, which will hit about the same time. Although David Wyss says that's not enough to trigger a recession, Ed Yardeni, chief economist at investment bank Deutsche Morgan Grenfell, Inc., predicts that there will be a 60 percent chance of a y2k-induced recession. He is threatening to up that percentage to 100 percent if no action is taken in a summer 1998 meeting of the G8 countries.⁵

In testimony before the United States House of Representatives Science Committee, a Gartner Group analyst estimated that less than 15 percent of enterprises have completed y2k planning. That means that 85 percent of the world's organizations will be beginning the largest information technology (IT) project they have ever attempted at roughly the same time.⁶

Capers Jones, a programmer productivity analyst and y2k expert with Software Productivity Research, says it will take 700,000 analyst-years to fix all of the software affected by the y2k dilemma. Even if time were available, a study from Information Technology Association of America shows there are not enough computer programmers and computer scientists to fill the 350,000 currently existing job vacancies.⁷

Bruce Hall, research director at Gartner Group, estimates that half of all organizations will not be able to update their systems in time for the year 2000. Thirty percent of mission-critical applications won't be ready, he predicts. Some companies that ignore the y2k problem will face lawsuits, bankruptcy, or go out of business.⁸

The estimated y2k price tags for large companies are now appearing in Securities and Exchange Commission (SEC) filings. Some of the larger y2k spenders are shown in table 1.2.⁹

TABLE 1.2

<i>Company</i>	<i>Spending Projections in Millions of Dollars</i>
Citicorp	\$600
General Motors	\$410–540
BankAmerica	\$380
AT&T	\$350
GTE	\$350
Chase Manhattan	\$300
Bell Atlantic	\$200–300
J. P. Morgan	\$250
Bankers Trust	\$180–230
Owens Corning	\$179

Some say Gartner Group's estimated \$600 billion worldwide price tag is not high enough. Paul Strassmann, technology consultant, says The Gartner Group, as well as the average business, underestimates the scope of the problem and does not take into account the embedded systems that are likely to fail. These include items such as global positioning satellites, building security systems, and logistics tracking systems, according to Strassmann. The \$600 billion figure also does not include:

1. The \$100 billion in litigation costs expected to arise from systems that aren't fixed in time
2. Differences in complexity of the code that need to be changed
3. Updates to test programs, environments, and tools
4. Database conversion costs
5. Integrated systems testing costs¹⁰

Strassmann bases his comments on a report written by Capers Jones, Chairman of Software Productivity Research, Inc. Jones estimates the worldwide y2k costs to be \$1.6 trillion. In the United States alone, his estimate is \$277 billion, which works out to \$989 for every U.S. citizen.¹¹

Regardless of which estimate will be closest to the final tally, solving the y2k dilemma will indisputably be the largest business incident of our lifetimes.

I realize I am starting off somewhat strong here. Some of you must be very skeptical. Common sense tells us that it just can't be that big of a problem—it's just too simple. But the y2k dilemma generates the type of problem that the more you know, the more you understand what a

nightmare it really is. Nigel Martin Jones of Data Dimensions was recently quoted in *Newsweek*: “There are two kinds of people: those who aren’t working on [a y2k project] and aren’t worried, and those who are working on it and are terrified.”¹²

The less you know about the y2k dilemma, the more unprepared you and your company will be.

It is an understatement to say that there has been some sensationalism surrounding this topic. In section I, “The Year 2000 Problem,” we will attempt to differentiate between events that will have a high probability of happening and events that will have a very low chance of happening within the y2k realm. Let’s take a look at what has already happened for starters.

- Five-year forecasting systems started gasping early in 1996.
- Credit cards with 00 expiration dates have been rejected by merchants’ card readers.
- A state prison computer ordered an early release of some inmates; it calculated that they had completed their sentences.¹³ They did not get out—guards overrode the system. Human controls as checkpoints will be important in high-risk systems like prison computers.
- Banks have seen glitches in their loan processing systems, in which loans can span decades.
- Perfectly good corned beef inventory (an oxymoron except to Brits?) was destroyed by a Marks and Spencer computer because the computer thought the corned beef was nearly a hundred years old.¹⁴

What is expected to happen in the future? Capers Jones is willing to put some probabilities on the following events:

- 70 percent probability of credit report errors
- 55 percent probability of a loss of electrical power for more than one day
- 35 percent chance of a loss of international telephone service
- 20 percent chance of a loss of local telephone service

- 20 percent chance of manufacturing shutdowns of more than one day
- 15 percent chance of bank account balance errors¹⁵

Exactly how did we get into this mess? Let's define the y2k dilemma.



A DEFINITION OF THE Y2k DILEMMA

My first knowledge of the y2k dilemma occurred in 1983, the first year I became a programmer. A very smart, senior programmer explained the dilemma for me. Because there was no room left on punched cards, and because storage costs were very expensive, the century portion of a date field was never stored in the computer. It was always assumed to be 19. When dates were stored in the computer, most of them were stored in a six-digit format, *MM-DD-YY*, where *MM* was month, *DD* was day, and *YY* was the last two digits of the year. For example, *060183* was the sixth month, first day, and eighty-third year. As programmers, we hard-coded tables (that is, we programmed fixed, unchanging values) inside the program. For example, *06* changed to *June*, or *83* changed to *1983* before we printed the values in a meaningful fashion on reports or screens.

This seemed innocuous at the time. That is, *06* was likely to remain *June* in my lifetime. But programs that used dates in calculations presented special issues. When subtracting a person's birthday from the current date to get age, for example, not all programmers used the full four-digit year for the computation. Those that used a two-digit year could receive negative results or just plain incorrect results.

You may have heard the story about the 104-year-old woman who received a notice to attend kindergarten. She was born in *93*, that is, 1893. But the computer didn't distinguish the century. I was born in 1956. In 2000, some computers will think I am – 56 years old. Does that mean I am not born yet? Will I still get a paycheck? These are a few of the tangled questions of the y2k dilemma.

In 1983, it seemed doubtful that the systems we were working on, some written in the 1960s, would make it to 1999. Nevertheless, we put a policy in place to fix selected programs for y2k. Any programs that we had to repair as part of other projects would also be repaired for the y2k problem. This would continue until we got closer to the problem.

Another procedure in this particular Fortune 500 shop was that there was a standard date subroutine that had to be used for all date validations and date computations. This centralized the issue: As long as the date routine could handle four-digit dates, the programming work was greatly minimized.

When I first learned about this problem, I remember thinking to myself: *Get out of this profession before 1999.* I didn't want to be "on call," responsible for any system of any significance during the last midnight in 1999. Well, anyway, so much for promises I can't keep.

What has happened since 1983? Not enough. Between programmers and their management, the proverbial ball got dropped, big time. In the mid-1980s, as punched cards disappeared and storage costs tumbled, the standard of storing two-digit years *should* have been reengineered. In some newer systems, the four-digit date was followed. But in most cases, the four-digit standard was never adopted. Why it wasn't is a question that will be pondered by historians in the technology industry for decades to come.

Although small businesses will not have the hefty price tag of the Fortune 500 companies, they will not be able to ignore the issue entirely. Just about every business will see some impact of this problem.

Not only will this problem invade most businesses, but it will affect every one of us personally.

CHAPTER 2: DON'T TAKE IT PERSONALLY, BUT. . .

So the y2k dilemma is just a business issue, right? What about the companies that you do business with? How will they be affected? To answer these questions, I called some of the businesses of which I am a customer. First, my credit card companies.



CREDIT CARDS

In 1997, merchants and the major credit-card companies struggled for a while to enable credit-card processing systems to accept cards with expiration dates in 2000 and beyond. First USA, Inc. recalled a few thousand cards with 00 expiration dates in early 1997. A credit union in Rye, New York, handled an overload of calls from Visa debit card holders whose cards with 00 expiration dates were rejected at merchant stores.¹ Credit cards with 00 expiration dates are now widely accepted without trouble.



OTHER EXPIRATION DATES

Credit cards aren't the only items that expire. Driver's licenses do, too. One state that usually issues driver's licenses that expire in four years has had to cut back to three-year expiration dates to work around the problem.²

What else can you think of that has expiration dates?

- Other bank cards besides credit cards, such as ATM cards and debit cards
- Equipment leases
- Magazine subscriptions
- Insurance policies
- Membership cards
- Licenses
 - Game (hunting, fishing)
 - Driver's

- Business
- Professional
- Warranty and maintenance contracts

The software and hardware that process these items must be checked for y2k compliance.



BREAD

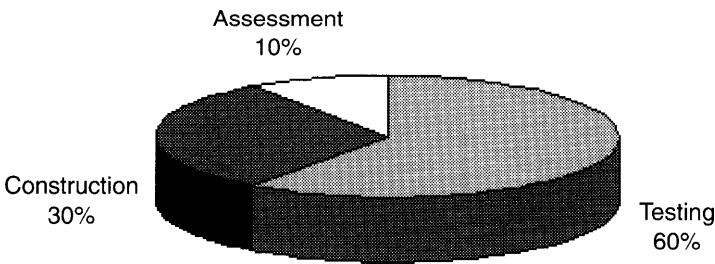
Chase Manhattan Bank has taken a leadership role in its industry. It disclosed in its annual report that it expects to spend up to \$300 million on its y2k effort to ensure that its \$trillion-a-day transaction level is not threatened. Although \$300 million sounds like a lot, Chase spends \$1.8 billion a year on technology, and therefore, the y2k project will gobble only 12 to13 percent of its total technology budget.³ The Chase company expects to fund the project from its existing technology budget.

Chase has been working on the project since 1995, but problems arose even earlier than that. The trading system that works with long-term bonds and the loan system are two examples of systems that had to be patched years ago.⁴

The project will be handled in three phases (see figure 2.1).

1. Assessment, which is expected to take 10 percent of the project effort, includes taking an inventory of systems.
2. Construction, the actual repair, will take 30 percent of the project.
3. Testing, which will take 60 percent of the project effort, is the largest phase. Chase will perform complete systems testing to insure that code works together as planned.⁵

FIGURE 2.1: Y2K PROJECT PHASES AT CHASE



Chase set up a y2k Program Office, a kind of clearinghouse for all company y2k issues. This office will coordinate the entire project and will work in conjunction with the Information Technology Division and Chase's business units. A searchable Lotus Notes database will also be available later in 1998, for employees to peruse to learn which software products are compliant and which are not.⁶

Although most banks were on schedule with their y2k compliance projects according to early 1998 Federal Deposit Insurance Corporation (FDIC) reports, a few banks were not faring as well. Late in 1997, the FDIC issued three cease-and-desist orders against three Georgia banks owned by Putnam-Greene Financial Corporation. As of February 1998, the FDIC itself was eight months behind schedule in its y2k project.⁷

A small bank has a different set of problems from a large bank, because it might rely on an outside party to process its transactions. All banks, regardless of size, must evaluate every entity with which it exchanges electronic transactions in order to ensure continuity of transaction processing.



MORE BREAD—INVESTMENTS

I called T. Rowe Price, a mutual funds company. Rowena Itchon, a spokesperson for the company, said its systems are new, and the company would not have a big problem with y2k.

The securities industry, as a whole, is expected to spend \$6 billion on the problem.⁸ Companies in this group include stockbrokers, dealers, stock exchanges, brokerages, stock trading firms, investment banks, and clearinghouses. The Securities Industry Association has created a y2k committee to address the problem in this industry.⁹ About thirty institutions were expected to begin limited testing in July 1998, followed by more intensive testing in March 1999, and possibly September 1999, if needed.¹⁰

One investment company has had fun with y2k testing already. It deposited \$19 million in each of its customers' accounts from a wayward y2k correction. Unfortunately for the customers, the glitch was caught and corrected right away.

In early 1998, there was talk of declaring December 31, 1999, a holiday for trading purposes. As far as I know, this idea was rejected. After all,

December 31, 1999 will be a Friday, so they have all weekend to get ready.



HARD-COPY BREAD

Are you still banking the old-fashioned way, with hard-copy checks? Check out the form your bank uses. How is the date formatted (see figure 2.2)? Does the date line read “DATE_____” or “_____19___”?

FIGURE 2.2: FORMS WITH _____19___ SHOULD BE UPDATED

SANDRA L. SMITH
6457 CEDAR HOLLOW 214-248-8378
DALLAS, TX 75248

0559
88-7450/3131
253

PAY TO THE ORDER OF _____ \$ _____

SAVINGS OF AMERICA
A Division of Home Savings of America, Inc.
Core/Beltline Office 253
110 N. Cox, Richardson, TX 75080

DATE _____ 19___

_____ DOLLARS

MEMO _____

Most forms have been changed already, and I realize that this issue is trivial because we can simply cross out 19 and write 20. It won't hurt to check out your entire forms stock and look for date fields that need to be updated.

Intuit ships checks that will handle the century change, as shown in figure 2.3. If you look closely, you will see a single quotation mark just before the two-digit year. Could that be a harmless y2k bug? Intuit states that QuickBooks 5, which generated the check in figure 2.3, is y2k compliant.

The best solution is to bank online, where check forms are unnecessary. (Well, that's assuming the Internet and telecommunications equipment are y2k compliant.) One consulting firm is advocating obtaining hard copies of statements of all of the investment and bank accounts that an individual owns.¹¹ This is a good idea, not only for y2k purposes, but in case of fire, Internal Revenue Service (IRS) audits, or software (or people) glitches in general. Of twenty-one industries ranked by Gartner Group, banking, investment services, and insurance are three of the best-prepared industries.¹²

FIGURE 2.3: THE FORM IS Y2k COMPLIANT—NOW WHAT ABOUT THE SOFTWARE?

BROWN HARRISON ASSOCIATES
123 MAIN STREET
PEORIA, IL 60652
(312) 555-1211

FIRST STATE BANK
PEORIA, ILLINOIS 60655
62-98/9967

1022

2/29/00

PAY TO THE ORDER OF anyone \$ **12.00

Twelve and 00/100***** DOLLARS

anyone

Check Style Shown:
Classic Voucher - Blue 1-Part
(Item #785)

MEMO _____

SAMPLE-VOID

⑈001022⑈ ⑆987654320⑆ 56789⑈ 1234⑈

WATER

My water company is the Dallas Water Utilities, which is a department of the city of Dallas. Chuck Mumm, Manager of Microsystems, is in charge of the department's y2k effort for everything but the mainframe. That includes everything from personal computers (PCs) to pipes all over the city and the process control systems that go with them. His team is in the inventory and evaluation stage, in which systems are being identified and prioritized as to criticality.

I asked him what the worst-case scenario could be (but probably won't be, because it will be fixed by 2000). Chuck said the water pressure might not be reliable if enough control stations fail. As the demand for water changes throughout the day, the pump stations located throughout the city must be altered to meet the changes in demand. This requires turning on or shutting off pumps in certain areas. A lot of these stations are not fully automated and are closely supervised by Water Department staff. So, even if many of the chips in the pump systems fail, enough people can override the control system so that it won't fail.

Researching the problem seemed formidable. Most of the systems were produced in the 1960s and 1970s by special contract. These manufacturers now have to be tracked down and questioned about the compliance of the components they sold to the Water Department. If the manufacturers can be tracked down and it is determined that the part needs correction, the next step is to determine the cost of the repair and who should pay for it. Once this price tag has been totaled, it will be compared to the cost of

scrapping and replacing the whole system with a newer, sleeker model with more functionality.



ELECTRIC

My electric company is TU Electric. H. B. Keating is Vice President of Information Technology there. TU Electric has the following y2k initiatives:

- IT, which consists of the mainframe and application software. This encompasses systems such as billing and payroll.
- Corporate, which includes the power plants and their control systems.

The IT initiative started in May 1996, and the initial assessment was completed in August of the same year. The project is expected to be completed by the end of 1998.

The corporate initiative has recently been started, and an assessment is under way. I asked Mr. Keating what the worst-case scenario could be (but that won't occur because the corporation is on top of it). Keating said that plants could fail. The power could go out. The plants may not be able to deliver electricity to homes and businesses. I asked whether there might be any safety issues. The answer is no, because many of the control systems were PC-driven.

An electric utility industry analyst, Richard Cowles, with TAVA/R.W. Beck LLC, said he thought the Nuclear Regulatory Commission (NRC) would have to shut down about 10 percent of the country's nuclear power plants due to lack of y2k readiness. The agency recently ordered the 104 nuclear power plants' owners to report on compliance and expects to audit a sampling of plants that are least prepared. The due date for compliance is July 1999.¹³

A 1997 Internet post on *CIO Magazine's Year 2000 Online Conference* yielded a note from a project engineer in charge of the y2k project for two nuclear plants in Virginia. He was looking for "information on how to evaluate, test or modify . . . system process and control devices."

Embedded systems such as these are tricky and expensive to test.

Simulation environments can be created for testing, but they are almost never carbon copies. Almost always, the manufacturer must be involved to disclose original specifications or other information to learn about a

device's compliance. For the sake of my CPA friends living in Virginia, I hope the engineer finds what he is looking for.



PHONES

Telephone companies must not only fix their billing and accounting systems, they must also correct problems in network management and network switching systems. AT&T Corporation has 500 million lines of code to review. MCI Communications Corporation has hired Data Dimensions, Inc. to perform its assessment. Sprint has 100 million lines of code and has six full-time and twenty part-time employees on the project. In addition, it has hired an outside firm to complete an assessment.¹⁴



CARS AND PLANES

I called Honda to find out whether my car will not start on January 1, 2000, as *Newsweek* suggested.¹⁵ I'm glad to report that my Honda's, computers keep track of the odometer reading instead of the date. So chances are high that my car will start on January 1. Unless, of course, I have pulled an "all-nighter" at a certain once-in-a-century party and left my lights on when I pulled up to park the night before. Cars with fancier features such as a "smart-start" ignition system may not fare as well.

Aircraft manufacturers such as Boeing rely on hours of operation instead of dates, so the y2k limitation does not apply to aircraft engines.¹⁶

Although airplane engines may not have a problem, navigation systems and some cockpit components could. Navigation systems will not have a y2k problem, however. They will have their own date problem. Global positioning systems (GPS) will have a problem at midnight, August 21, 1999. That's because satellites have a 1,024-week window in their software that begins on January 6, 1980, and ends on August 22, 1999. Satellites measure time in weeks and seconds instead of the usual date format. Newer GPS units will already be corrected for the problem, but if you have an older unit, you should contact the manufacturer for a firmware update.^{17,18} GPS units are popular with pilots, airlines, rental car companies, campers, sailors, and hobbyists.

The Federal Aviation Administration's (FAA) y2k project has been criticized heavily in the general press. The nation's air traffic control system runs on forty noncompliant IBM 3083 mainframes, vintage 1970s. FAA

employees must migrate the system to a new hardware platform and are reliant upon their internal programmers as well as contract vendors for y2k software modifications. In the spring of 1998, the FAA hired Coopers & Lybrand and created a centralized y2k program office to accelerate the efforts.



VCRs AND OTHER APPLIANCES

An easy way to tell whether a household appliance could have a problem is to check whether it holds a date in the first place. My VCR is on a seven-day, twenty-four-hour cycle, so it is not smart enough to know what year it is. It shouldn't care whether 2000 comes and goes. (This is not an all-inclusive test some devices fail even if the date cannot be entered anywhere.)

Only a small percentage of appliances, mostly the more sophisticated ones, will have a y2k problem. Even if they do, most of these failures will fall into the nuisance category. If you are unsure of the status of your appliance and its operation is critical, you should contact the manufacturer for more information.



HEALTH CARE

It might be a good idea to try not to get sick in 2000. Of twenty-one industries ranked by Gartner Group, health care, education, and semiconductors were the three lowest-ranked industries.¹⁹ Many hospitals are far behind in their y2k efforts. The industry has seen an increase in mergers and acquisitions and is suffering from tight budgets in the managed care arena. Worse, many biomedical equipment vendors have not spent much time on the y2k issue.²⁰ Some medical products can generate harm if they fail from y2k implications. One manufacturer has already recalled some heart defibrillators that could not handle the century date change. The device's internal clock tracked the time since the last maintenance. If maintenance is not done in time, the defibrillators stop working.²¹

The top five pacemaker manufacturers were interviewed in a U.S. Veteran's Administration project. Four companies were working on the problem; one would have the fix in 1997, one in 1998, and two before 2000. One company refused to acknowledge the problem.²²

The greatest risks involve medical equipment that is old and still in use. Companies that use medical devices can test their equipment by resetting the clock, preferably when the device is detached from patients.²³ They can also contact the vendor and question them about compliance. Hospitals should correct medical equipment before correcting their billing systems, but these suggested priorities have not been followed in many cases.



GOVERNMENT SERVICES

Most government agencies and many state and local agencies will not be completely prepared for 2000. Gartner Group says 30 percent of the federal government's systems will not be compliant by the deadline. The states are in worse shape. In 1996, a survey by the National Association of State Information Resource Executives (NASIRE) disclosed that 40 percent of states have not started y2k projects.²⁴ To find out how your state is doing, check out the NASIRE's y2k page at <http://www.nasire.org/year2000/index.html>.

There are a few bright stars. The Social Security Administration started to address the y2k issue in 1989 and is a model for the rest of the government. The State of Nebraska has a plan. But in general, the outlook is bleak. In a House of Representatives report, Representative Maloney (D-NY) is quoted as saying, "Systems that deliver services to individuals will not work, and those services will not be delivered. Checks will not arrive on time."²⁵ In February, a General Accounting Office (GAO) director testified that "virtually every citizen" could be affected and that "every federal agency is at risk of system failures."²⁶

Even stars can have a bad day. In fall 1997, the Social Security Administration discovered 33 million lines of code in interfaces with state disability systems that it had overlooked as part of its original plan. Since they were so far ahead with their y2k plans, managers spent little time developing contingency plans to address events such as the overlooked code. Now those mission-critical systems that pay out \$386 billion in annual benefits may not be remediated in time.²⁷

In spite of all of this, the Social Security Administration was the only government agency to earn an A+ in a May 15, 1998, grading by a House subcommittee. Overall, the government earned an F in overall y2k preparedness. The scores of selected agencies are shown in table 2.1.²⁸

TABLE 2.1

<i>Agency</i>	<i>May 15, 1998 Y2k-Readiness Grade</i>
Social Security Administration	A+
General Service Administration	A-
Federal Emergency Management Agency	A-
National Science Foundation	A-
Department of Commerce	B
Small Business Administration	B
National Space and Aeronautics Administration	B
Nuclear Regulatory Commission	B
Department of the Treasury	C
Department of Housing and Urban Development	C
Department of Labor	C
Department of Veterans Affairs	C
Office of Personnel Management	C-
Department of the Interior	C-
Department of Agriculture	D
Department of Defense	D
Department of Justice	D
Department of Education	D
Environmental Protection Agency	F
Department of State	F
Department of Health and Human Services	F
Department of Energy	F
Department of Transportation	F
Agency for International Development	F

Flat tax proponents are delighted with the opportunity that the y2k dilemma brings to the IRS. Although the IRS is fairly good at patching old systems for tax law changes, they face a multitude of challenges through 1999. Taxpayers and Congress are demanding an overhaul to create a friendlier agency, the agency's y2k project architect resigned early in 1998, and IRS programmers are leaving for higher pay in the corporate sector. Consider adjusting your withholding in 1999 (and advising your clients likewise) so that the IRS doesn't owe you anything come April 2000.

In February, 1998, President Clinton issued an Executive Order for government agencies to be y2k compliant. He also appointed a Council on the Year 2000 Conversion, which is headed by John Koskinen who has a background in crisis management.²⁹

The United States Senate recently created a special committee to handle y2k issues. Senator Bob Bennett (R-Utah) chairs the new Senate Special Committee on the Year 2000 Technology Problem.³⁰

CHAPTER 3:

STRICTLY BUSINESS

In this chapter, I will present how business systems will be affected, beginning with PCs. I'll discuss packaged software and legacy systems. Finally, I'll present a sample of systems that you might not expect to have y2k problems.



PCs

I decided not to put it off any longer. I flipped on my Toshiba laptop, which I purchased in March 1996, set the date to December 31, 1999, and set the time to 11:58 p.m. Then I turned it off and waited five minutes. Read on to see what happened.

The press coverage originally had me very scared about my PCs. Will I have to replace them? Will they be no more good than “boat anchors and doorstops”¹ after the millennium? Most likely, no. Most of the PCs that I have encountered will work just fine past 1999.

The problem with PCs and the year 2000 is minor compared to the larger impact of y2k. If you work in a small business with only a few PCs, you may be able to get by with doing nothing until January 1, 2000. Then, if the date is wrong, just change it. If you want to be a little more prepared, or if your system will be running mission-critical date applications in your absence just after the millennium, or if you are in charge of a large network, you may need to be a little more educated. Read on.



A LITTLE TEST

If you are not sure about whether your PC is y2k compliant, you can try the same test on your PC that I ran on my Toshiba laptop. The test should be done for each PC that you intend to keep after 2000. It will confirm whether the hardware and operating system are compliant or whether additional work must be done to ensure compliance.

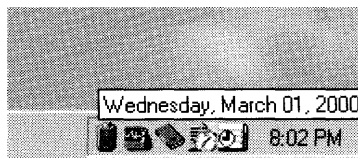
Before you start the test, take note of the following three warnings.

1. Back up your hard drive before starting this test.

2. If you have any software that expires or stops running as of a particular date, you should not perform this test. This test could render the software inoperable. Many tax research libraries have expiration dates on their software, so don't say I didn't warn you.
3. If you have any software that will start when the computer does, disable it. For example, Symantec's Act! runs in my startup group. I keep my schedule on it and have a feature enabled that moves all incomplete tasks to the current date. If you set your PC ahead to 2000, what will happen to your scheduled tasks? Your entire schedule will be obliterated, and the computer will anticipate that you are going to be extra busy on January first.

When I turned my laptop back on after five minutes, I checked the system date for a correct century display. It read January 1, 2000. It passed. If your computer passed this part of the test, then you should test February 28, 2000, and December 31, 2000, as well. Four percent of PCs are expected to have leap-year problems.² Also, be sure to test *all* PCs, including those that you suspect have the same motherboard version. Tests have shown that PCs with the same motherboard version can yield different results. If you have hundreds of PCs, you'll want to get some automated tools to speed the process. Success with all three tests means you can skip past the next few steps and move to the software part of the test (see figure 3-1).

FIGURE 3.1: BOTTOM CORNER OF WINDOWS 95 SCREEN AFTER CHECKING LEAP DAY, 2000



If your system does not display the century correctly, you can try to set the date at this time. If you can set the date, then you have passed another hurdle. Now, turn the machine off and on again to see whether the system will hold the date. Be sure you do a full power off/power on; a system restart is not the same thing.

If the system will hold the date, then your system has passed this part of the test. Most PCs in operation today will fall into this category. They will

not need any hardware correction. However, you will have to remember to reset the system date on January 1, 2000.

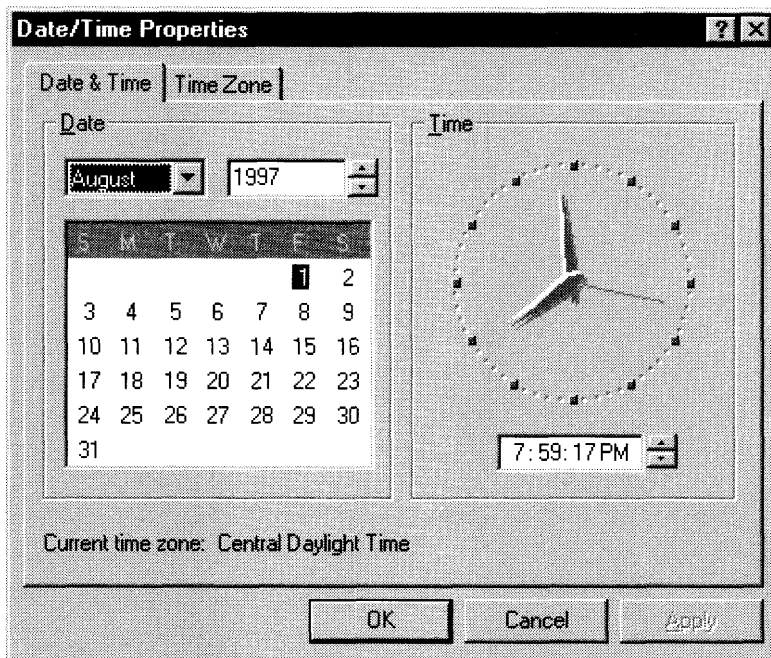
In early 1998, I bought a 1994 Toshiba T4400SX laptop to use on a work trip to a remote part of Nepal. I tested it for compliance right there in the store before I purchased it. When I turned the machine back on after resetting the date, it read 1-4-80. When I entered 1-1-2000, it worked just fine. I have to remember to set the date on January 1, 2000, if I still have this computer by then.



RESETTING THE DATE

To reset the date in Windows 95, double-click on the time display in the tray portion of your task bar. The Date/Time Properties screen will appear (see figure 3.2).

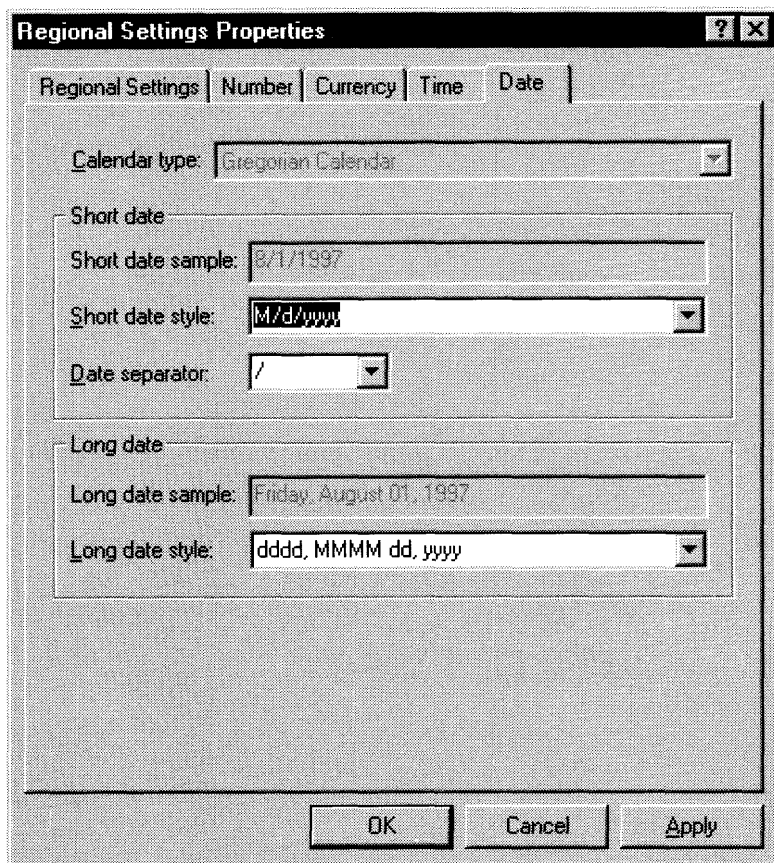
FIGURE 3.2: THE DATE/TIME PROPERTIES WINDOW



Simply click on the portion of the date or time that you wish to change and enter the figure or use the up and down arrows to increase or decrease the values. Click **OK** or **Apply** and the date and time will be changed.

Windows 95 has another feature related to dates in the Regional Settings icon in the Control Panel window. Here, you can select a short and long date display style with either a two-digit year or a four-digit year (see figure 3.3).

FIGURE 3.3: WINDOWS 95 DATE STYLES



At the DOS (disk operating system) prompt on a DOS PC, you can type **DATE** and the system will return the current date and prompt you for the new date in MM-DD-YY format. You can then enter 01-01-2000. Be sure to enter the year as a four-digit number; if you enter 01-01-00, it says it's an invalid date. (Note that this is a change from what I said in the first edition of this book.) You can also set the time in a similar fashion via the **TIME** command.



THE TECHNICAL EXPLANATION

An AT (Advanced Technology) compatible PC uses the real-time clock (RTC) to keep track of date and time. The RTC is a battery-driven piece of hardware that runs even when the machine is turned off. It is limited in space, however, and there are only two digits to store the year. It is complemented, however, with an extra digit in the complementary metal-oxide semiconductor (CMOS), which was allocated in later PC designs to store the century.³

When the computer is turned on, the RTC and the CMOS together come up with the date value when the BIOS (basic input/output system) calls for it. The DOS date command can be used to reset the value needed; for example, if the battery is running low, if a time zone change occurs, or if a similar issue occurs to render the date value incorrect.⁴

When it becomes 2000, the computer will interpret the year as *00* and the century as *19*. The problem is then compounded within the operating system limitations. DOS recognizes the beginning of time as it knows it as 1-1-80. DOS can recognize dates from 1-1-80 to 12-31-2099. DOS edits the date value for out-of-range conditions. Year must be 1980 to 2099; month must be from 1 to 12; and day must be from 1 to 31. A date of 1-1-1900 will be returned the first time the computer is turned on after 2000, and this date is out of range for DOS. DOS returns the arbitrary date 1-4-80 when it encounters an out-of-range date condition. Many of you have probably seen this value on your computers. DOS sets the date to arbitrary values when other conditions arise. If you have ever lost battery power or replaced your battery, your PC date would be set to 1-1-80. If an invalid binary-coded-decimal (BCD) code is encountered, the date would become 1-3-80.⁵ This is a type of error message display, albeit cryptic.



OPTIONS FOR PCs

There are many options for PCs. In many cases, including the following, you will not have to throw away your old PC.

1. If your PC is networked, it may be able to retrieve the date from the network. One company with a large Novell network plans to run a script that will change the date on its

PCs. Any plan should be tested thoroughly, especially if the PCs are running mission-critical operations.

2. If you had to change the date but your system will hold the date, decide whether you want to change the date manually. If you do, you just have to remember to do it on 1/1/00. Suppose you want the computer to roll over automatically, or your PC will be running mission-critical applications just after the millennium and you cannot physically be there to monitor the jobs, or your system will not hold the date, then you should do one of the following steps.
 - a. If the PC's BIOS can be modified, you can obtain a BIOS patch from the hardware vendor. Updating a BIOS is called flashing the BIOS; you may see the phrase "flash the BIOS" in conjunction with PC makers' instructions to correct for the y2k problem.
 - b. If your BIOS cannot be corrected with software, you may be able to obtain a hardware part that will fix the problem.
 - c. There are some companies producing utilities that will update the date or check your hardware for compliance. NSTL (which doesn't stand for anything, I was told by an employee of the company) offers a tool at http://www.nstl.com/html/yrmark_2000.html. SBT, at <http://www.sbt.com>, also has a tool. The RightTime Company offers yet another tool at <http://www.righttime.com>. Please check out any vendor's promises very thoroughly; I am listing these products for completeness, not to endorse them.
3. If the PC fails to hold the date after trying all of the above, it's time to do the following:
 - Run only non-date driven applications from this PC.
 - Make it a print server.
 - Give it to your kids or a kids' group.

If your employees are still using XTs, you should be ashamed of yourself. Although I couldn't find one to test, I do not think they recognize the year 2000. Whether they do or not, trust me: it's time to junk them and get Pentiums.

There are several good sources of information from hardware vendors (see figures 3.4, 3.5, 3.6). A few of the hardware vendors' Web pages are listed below for your convenience:

- Acer—<http://www.acer.com/aac/support/articles/yr2000.htm>
- Compaq—<http://www.compaq.com>, enter *year 2000* in the search engine
- Dell—<http://www.dell.com/year2000>
- Gateway—<http://www.gw2k.com>
- Hewlett-Packard (HP)—<http://www.hp.com/cposupport/eschome.html>
- IBM—<http://www.ibm.com/IBM/year2000/>
- Micron—<http://www.micronpc.com/about/year2000/>

In addition, the following y2k Web sites list vendors' positions on particular PC models:

- The State of Washington at <http://www.wa.gov/dis/2000/compliance/menu1.htm>

FIGURE 3.4: DELL'S Y2K PAPER

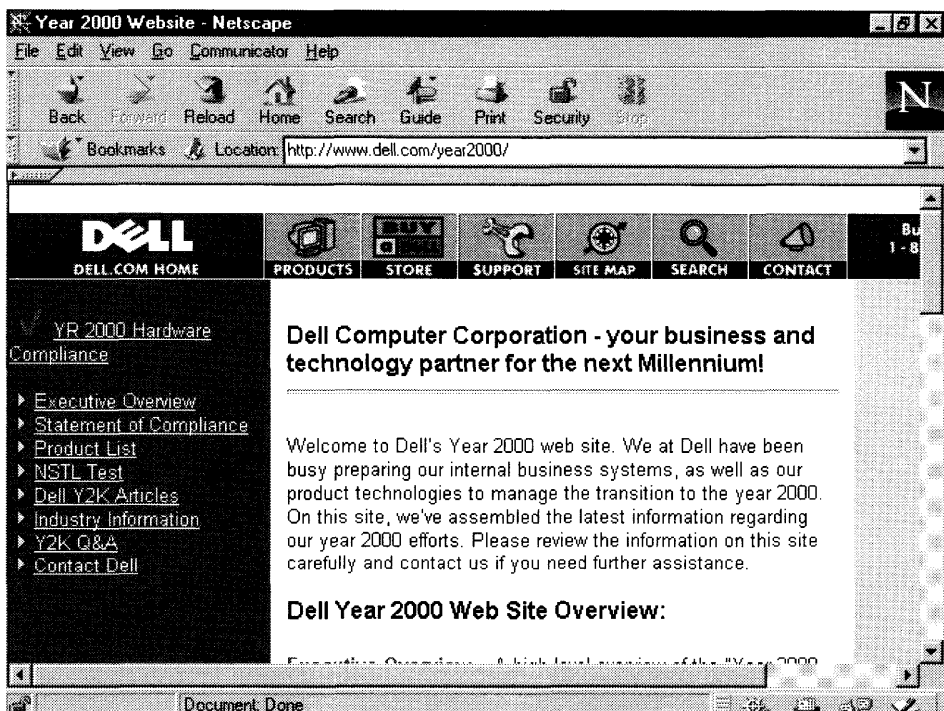
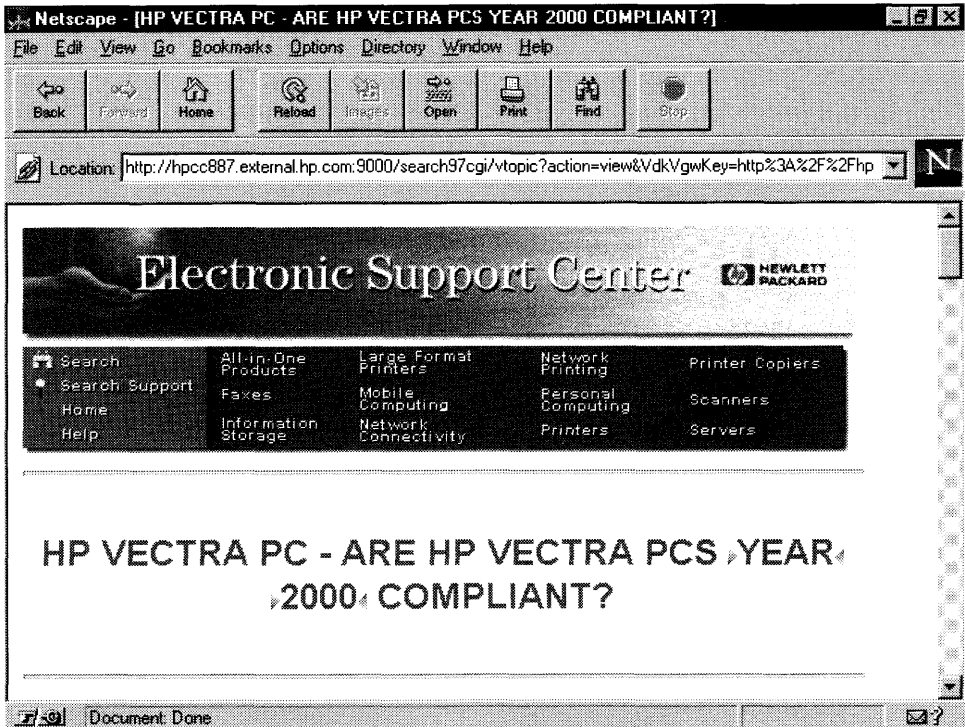


FIGURE 3.5: A SEARCH AT HEWLETT-PACKARD'S SITE YIELDS Y2K INFORMATION

- The Computer Information Centre, <http://www.compinfo.co.uk/y2k/manufpos.htm>



MISCELLANEOUS HARDWARE DEVICES

The y2k dilemma does not stop with PCs and laptops. In a network, devices such as routers, multiplexers, packet switches, channel extenders, remote access devices, and so forth, should be inventoried and checked. Many individuals have purchased personal digital assistants (PDAs) in the last few years. These devices, along with sophisticated calculators, like the popular HP-12C, must be checked for compliance. Other handheld devices, such as ones used for inventory control or point-of-sale ordering, must be considered as well. The best way to check all of these devices is to contact the manufacturer or visit the Web site and search for y2k instructions.

FIGURE 3.6: THE STATE OF WASHINGTON LISTS VENDORS' Y2k COMPLIANCE LETTERS

Name	Last modified	Size	Description
Parent Directory			
0000.xls	19-Nov-96 08:33	5K	
all.xls	20-May-97 13:46	1K	
all.xls	02-May-97 10:05	1K	
apple.xls	19-May-97 10:41	7K	
att.xls	14-Mar-97 10:30	1K	
awad.xls	04-May-97 10:05	1K	
comp.xls	30-Apr-97 12:29	8K	
ct.xls	24-Jul-96 09:58	1K	
dec.xls	14-Mar-97 12:45	1K	
del.xls	23-Apr-97 13:38	1K	
dt.xls	19-May-97 07:49	1K	
edward.xls	03-Mar-97 10:34	9K	
edward.xls	22-Aug-97 08:56	7K	
edward.xls	01-Jul-97 14:28	1K	
gmk.xls	22-Aug-97 08:49	1K	
hewlett.xls	23-Jul-96 07:44	3K	
hp.xls	24-Jul-96 10:37	3K	
ibm.xls	18-Mar-97 09:14	1K	

SOFTWARE IS A BIGGER ISSUE

Once you have determined the status of your hardware, your software must be analyzed. The heart of the y2k problem occurs in software. Each software package that you run on your desktop, including the operating system, should be checked to determine how it will handle the y2k dilemma. In this section, I will list some of the more common desktop software packages and their status. Software packages that do not use dates extensively, such as word processing programs, graphics, and desktop publishing, will not be mentioned because their problems are generally nonexistent or very minor.

Operating Systems

Apple boasts that MacOS (Macintosh Operating System) has been compliant since its inception in 1984. It correctly recognizes years from 1920 to 2019.⁶

OS/2 (Operating system/2) Warp has included a patch to roll the date over correctly to the new century so that its users do not have to do anything.⁷

Windows 95 requires a patch to correct for minor problems with the *DIR* and *DATE* commands. The patch is available free at <http://www.microsoft.com/year2000/>.

Windows 3.1 requires a patch to correct file manager problems where a 00 year shows as :0. The patch is available at <http://www.microsoft.com/year2000/>.

Windows NT 4.0 Workstation has minor y2k-compliance issues according to Microsoft. The User Manager does not recognize leap year, and there is a y2k problem with the Find files command. Details plus the patches are available at <http://www.microsoft.com/year2000/>.

Network operating systems deserve a few sentences, too. Certain versions of Novell Netware systems require patches in the operating systems and the clients. Go to www.novell.com/p2000/patches.html for the details. Microsoft NT Server requires patches as well, available at <http://www.microsoft.com/year2000/>.

Utilities

It's easy to forget utility software such as antivirus or backup utilities, but all of these packages should be listed in inventory and determined whether they are compliant. Symantec's Norton Antivirus, for example, is not compliant. Two y2k-related lawsuits regarding that product has been filed against Symantec. For more information on lawsuits, see Chapter 9, "Legal Issues."

Spreadsheets

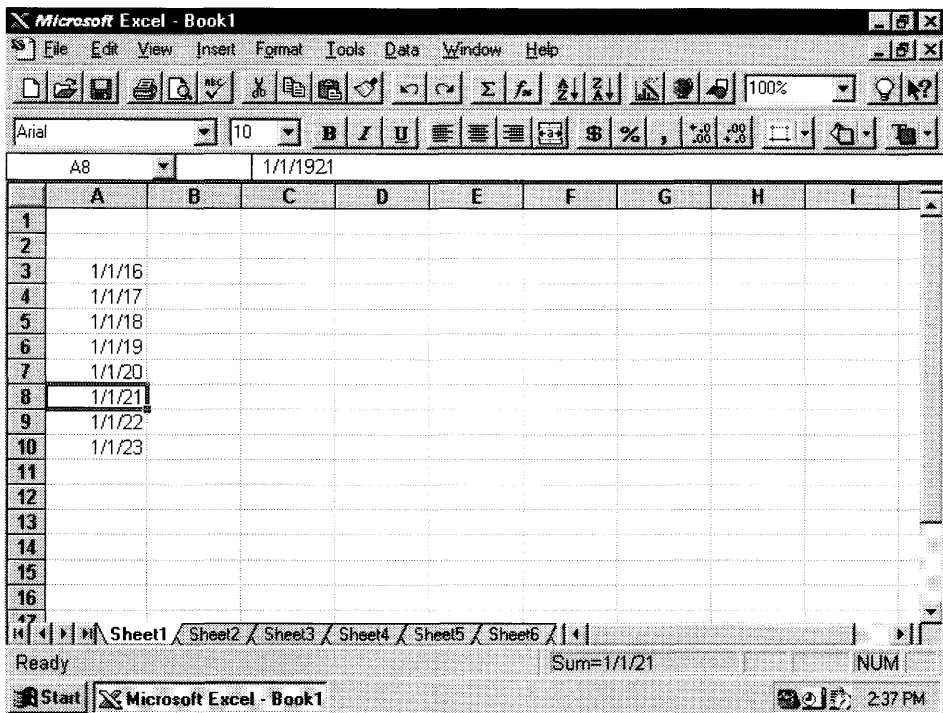
Most of the popular spreadsheet packages handle the year 2000, but on their own terms. It's a good idea to understand the assumptions for each package when using twenty-first century dates. Entering the year as four digits instead of two is probably the safest method. Most of the packages use the windowing method (explained in chapter 18, "Date Modification Options"), but each one uses a different pivot point. A program that uses a pivot point of 50, for example, interprets 00 to 50 as 2000 to 2050 and 51 to 99 as 1951 to 1999. The problem is that the pivot point varies for each software package and even for versions of the same software package.

In Microsoft Excel Version 7 for Windows 95, any year entered as a two-digit format after 2019 reverts back to the twentieth century. For example, if you entered the following dates in cells:

- 1-1-18—Excel translates to 1-1-2018
- 1-1-19—Excel translates to 1-1-2019
- 1-1-20—Excel translates to 1-1-1920
- 1-1-21—Excel translates to 1-1-1921

Microsoft Excel 97 in Office 97 has the same problem but at 2029. In either version, if you enter the years in four-digit formats, the date is not a problem until 2078. The next major version of Excel will extend the 2078 limitation to 9999 (see figure 3.7).⁸

FIGURE 3.7: MICROSOFT EXCEL AND THE Y2K DILEMMA



The different pivot points could cause problems with users that trade files between different versions of Excel. In Excel version 7, a year value of 25 is interpreted as 1925. If you open that spreadsheet in Office 97, the value becomes 2025 and could cause calculation errors. On a pension spreadsheet I saw recently, a woman whose birth year was 1925 had earned a sizable pension, but the spreadsheet showed her owing the pension several thousand dollars. Excel thought she was born in 2021 instead.

Entering dates with month and year only could cause problems in 2001. For example, suppose you enter *Dec 99*, *Dec 00*, and *Dec 01*. Excel interprets the first two as expected, but the third one is assumed to mean 12-01-YY, where YY is the current year. Since *01* can be interpreted as a valid day, the computer interprets it to be a day and not a year.⁹ Any date format that leads to ambiguous interpretation should be avoided.

Corel Quattro Pro 8 (and also Corel Quattro Pro 7 for Windows NT) assumes a pivot point of 50 (which has become, by the way, the default pivot point for mainframe programming changes). Entering a two-digit date for 2000 and beyond will work up to 2050. A date of *51* will be translated as 1951, but a date of *50* will be translated to 2050. In Corel Quattro Pro 7 for Windows 95, all two-digit date formats assume the century 19, so you must use the four-digit date format when entering dates after 1999. You can enter dates up to 3199 using a four-digit date format in any of the Quattro Pro packages.¹⁰

All releases of Lotus 1-2-3 accept two-digit and four-digit years and assume that if you enter two-digit years, you mean 19 for the century. Even if you are still running DOS 2.4 version of Lotus 1-2-3, you can key in values for the next century, but you must key in the value as a number of years past 1900. So, if you enter *64*, you'll get 1964, and if you enter *115*, you will get 2015.¹¹

By the time we reach 2001, the format MM-DD-YY will be very difficult to interpret. Does 03-02-01 mean March 2, 2001, February 3, 2001 (Europe), or February 1, 2003 (military)? Report and screen displays of dates should be designed to avoid ambiguity, whether that means printing four-digit years or printing the format (MM-DD-YY) that is assumed.

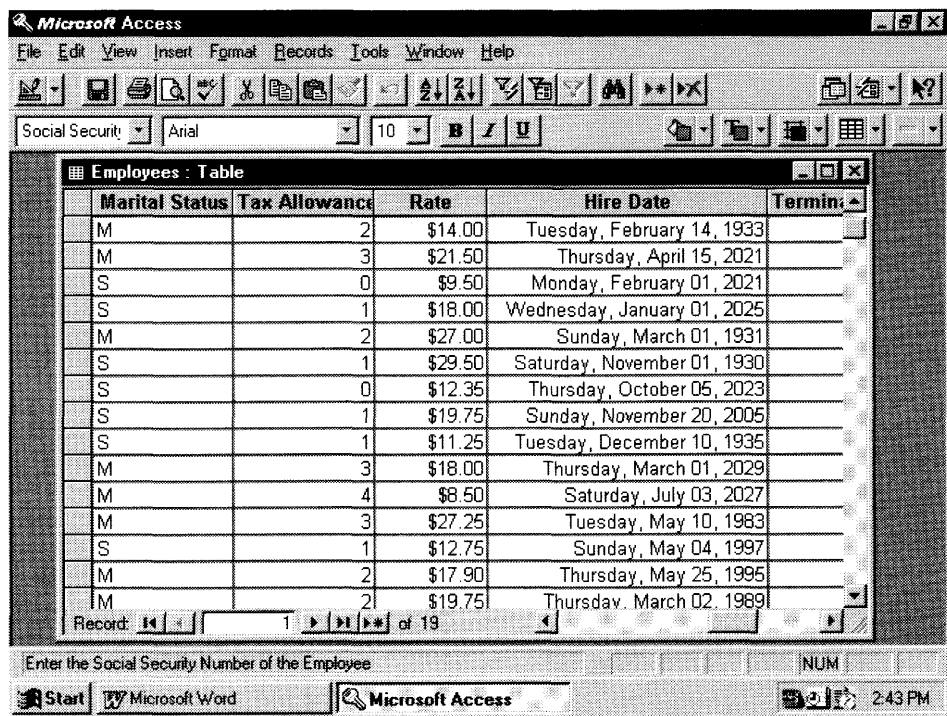
Two great sources for more information about spreadsheets are:

1. Microsoft's Year 2000 Resource Center, at <http://www.microsoft.com/year2000>. (Choose Products, then Excel.)
2. A Web page called "Year 2000 and Spreadsheets," by Patrick O'Beirne (Find it at <http://homepages.iol.ie/pobeirne/y2ksprds.htm>. I found this site while at Herbert B. Elliott, Jr.'s Mother of All Year 2000 Sites, at <http://pw2.netcom.com/~helliott/00.htm>.)

Databases

According to Microsoft, Microsoft Access 7 for Windows 95 interprets all two-digit years as being in the twentieth century. After 1999, you must key in the four-digit year. However, my screen print shows that when values from 00 to 29 were entered, 20 was assumed instead of 19. This is how Access 97 was documented to work. Access 97 in Office 97 assumes 19 with two-digit years after 2029 and assumes 20 with two-digit years prior to 2030 (see figure 3.8).¹²

FIGURE 3.8: MICROSOFT ACCESS MAKES ASSUMPTIONS ABOUT TWO-DIGIT YEARS



Borland Paradox 7 accepts four-digit years up to 9999, but if you key in a two-digit year, it will assume you mean 19.¹³

Lotus Approach assumes when you enter 00 to 29, you mean 20 for the century, and when you enter 30 to 99, you mean 19 for the century, when entering a two-digit year format. You can key in the four-digit year any time.¹⁴

Databases that have been designed by either professionals or users must be analyzed. It is good form to expand the date fields in the database layouts.

Any corresponding code that works with the dates should also be reviewed and changed, if necessary.

The information presented above was gathered directly from the vendors or from spokespersons. It has not been independently verified. It is a good idea to test any claims of y2k compliance well before the actual date change.

Since there is no standard definition of *y2k compliant*, it is prudent to dig deeper into a vendor's meaning of the term. For example, your idea of y2k compliance may be that you can enter two-digit years and the system will assume the correct year. The vendor's idea of y2k compliance could be that you can enter four-digit years and the computer will accept them. If your company has a tremendous volume of dates for data entry, the difference between keying two digits and four digits is significant.

Files, such as spreadsheets, databases, and macros, must be remediated or corrected for the y2k problem. This means taking an inventory, identifying the files that need to be changed, training the employees who will make the changes, and changing them. Any employee who creates these files should be trained to create y2k-compliant files. Because of the high probability of reinfection, training should be constantly reinforced until it becomes habit. Many firms are conducting periodic audits after remediation has been completed to ensure these files remain compliant.

For a CPA firm or any accounting department, the volume of current and historical files can be enormous. This step should not be taken lightly if there is any exposure from miscalculations in spreadsheets, databases, and the like. Adequate time should be included in the plan for remediation of these files and the associated training.



ACCOUNTING SYSTEMS

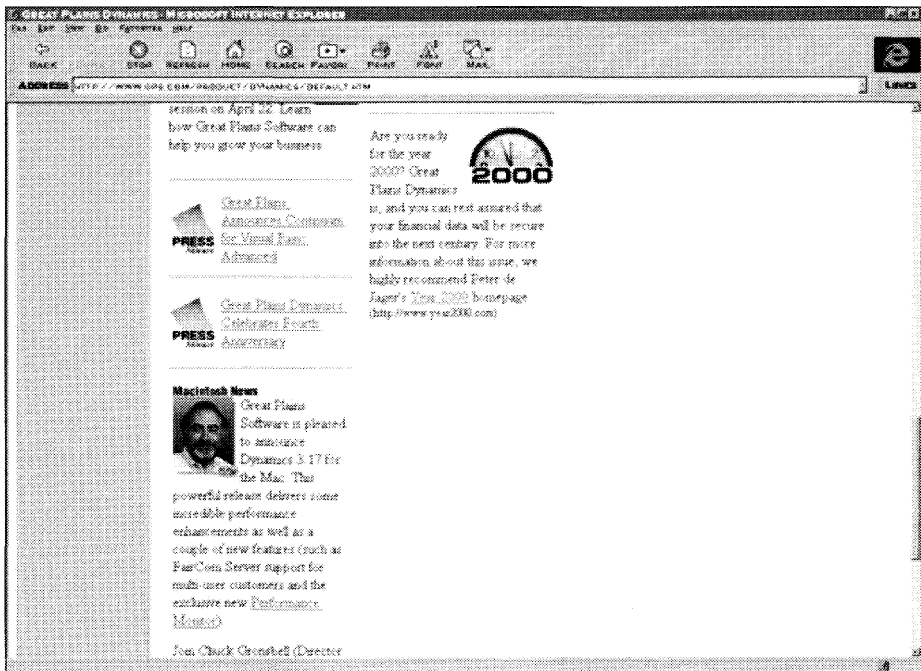
Some accounting systems vendors are on top of the y2k issue and others are not. Vendors, such as Great Plains, that have had y2k-compliant products for a while, can boast this fact in their marketing literature. Other companies are quietly posting upgrade and support plans on their Web sites. Here's a rundown of some of the accounting packages and their status on y2k compliance.

- According to Mark Pickens, CPA, of Nelson & Pickens, L.C. in Dallas, Business Works for Windows has been compliant since about version 7. Version 12 shipped in the fall of 1997,

and it is, of course, compliant as well. Nelson & Pickens is a value added reseller (VAR) for the product.

- DacEasy is y2k compliant beginning with version 8, according to its technical support service.
- Great Plains' Dynamics is y2k compliant. Its Web site has a white paper by Wayne Harding about the y2k issue (see figure 3.9).

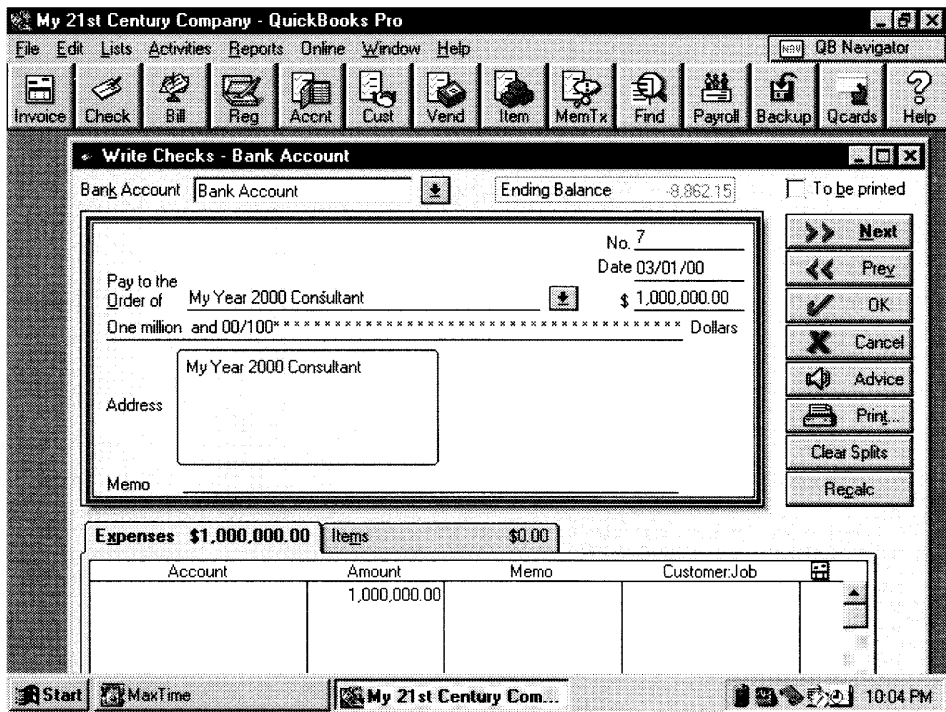
FIGURE 3.9: GREAT PLAINS DYNAMICS IS Y2K COMPLIANT



- Macola offers its Version 7.0 Progression Series for its y2k solution. A y2k-related lawsuit has been filed against Macola. For more information on the lawsuit, see chapter 9, "Legal Issues."
- Users of MAS90 will receive the y2k upgrade as part of their ClientCare support plan. It was expected to ship in the third quarter of 1997. Customers who were not part of the maintenance agreement were "strongly encouraged to subscribe" to avoid the upgrade fee.

- A spokesperson at PeachTree said that all of its Windows products were y2k compliant and its DOS product is y2k compliant beginning with version 10.
- Intuit's QuickBooks versions 4 and 5 are y2k compliant (see figure 3.10). Their personal finance package Quicken 98 is compliant. Note that this is a change from the information in the first edition of this book. Two y2k-related lawsuits have been filed against Intuit regarding Quicken 6. For more on lawsuits, see chapter 9, "Legal Issues."

FIGURE 3.10: QUICKBOOKS 5 IS Y2K COMPLIANT, PER INTUIT



- SBT Pro Series 3.2i lists "21st Century Support" in its product literature on the Web. A y2k-related lawsuit has been filed against SBT. For more information on the lawsuit, see chapter 9, "Legal Issues."
- Solomon III for Btrieve and Solomon IV for Windows are compliant according to the y2k page on Solomon's Web site.

Some vendors are using the term y2k ready instead of y2k compliant. A y2k ready software package is one that will handle four-digit dates in

multiple centuries as long as it is run on a y2k ready hardware platform and a y2k ready operating system, and any interfaces are also y2k ready. Y2k ready is a term that was devised for legal reasons to limit vendors' liability to what they can control, i.e., their own software package. For purposes of this book, we will use the term y2k compliant since we are referring to your internal company's goals, which is hopefully to be fully y2k compliant for all inventory components.

The information presented above was gathered directly from the vendors or from spokespersons. It has not been independently verified. It is a good idea to test any claims of y2k compliance well before the actual date change.



OTHER PACKAGED SOFTWARE

There are too many software packages to list, but we will list a few more categories of software that deserve attention.

- Contact managers
- Scheduling software
- Project management software
- Time and billing software

An easy way to research a product's y2k compliance is to view the Internet site, locate the search engine and key in the words *year 2000*. Most software vendors began posting their statements on their products' y2k compliance in late 1996.

In many ways, it is more difficult to take an inventory of PCs and network devices than of the mainframe and related components. You probably know how many mainframes you have, but do you know exactly how many PCs you have? Software is worse: Do you know what software packages users have downloaded from the Internet? (These should be counted in inventory, too.) The mainframe environment has been around for a longer period of time, and we have developed better change management tools for the mainframe than for the network. If your company has interfaces between the two environments, such as PC-to-mainframe uploads or mainframe-to-PC downloads, make sure your y2k plan ensures that data from one area do not contaminate data from the other. (Remember an old mainframe saying: The devil is in the interfaces.)¹⁵



LEGACY SYSTEMS

The bulk of the effort of a large company's y2k project will take place in mainframe systems called *legacy systems*. These systems consist of older technology and have not yet been replaced by newer technology such as client/server or intranet. Legacy systems have been written in a multitude of languages, the most common of which is COBOL. Many Fortune 500 companies are faced with the task of changing COBOL programs that are running on an IBM mainframe system.

A y2k change in one program is a relatively simple maintenance task. The legacy system scenario becomes complex quickly as additional details are considered, such as the age of the programs and the lack of standards in those days. Consider the following points about the COBOL environment in the 1970s and 1980s.

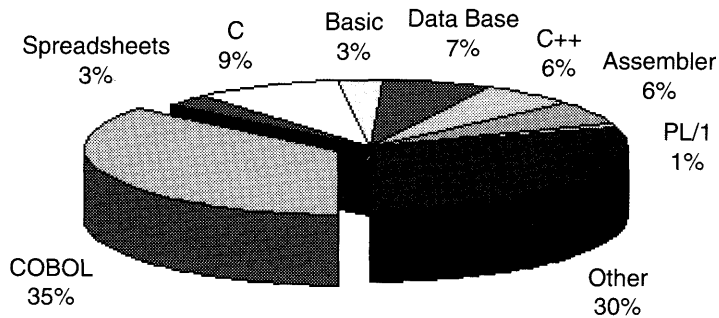
- COBOL programs are compiled with a piece of software called a compiler. The compiler creates the object code, which is machine-readable, and issues the machine instructions. This means there are both a source code version and an object code version of a program to retain in a library since program creation. Often, these two pieces of software get out of sync. A y2k team member must make a change to the source code of a program, then recompile. What if the compiled version that has been running all these years is not the same as the source version that is being changed? Surprises occur at run time; results are not as expected. Better yet, what if the source version is lost? This was very common with the earlier systems as good source management tools were not developed until later.
- Standards were not so well thought out in the 1970s when some of the COBOL code was written. Instead of the strict naming standards programmers must follow today, such as naming a date *TODAYS-DATE* or *EMPLOYEE-BIRTHDATE*, some programmers named dates after flowers and girlfriends. A y2k team member might find that *SALLIE*, *BUTTERCUP*, and *R2D2* are all date fields. This makes it hard to find all date instances.
- Today's programs are structured into logical sections or paragraphs, but yesterday's COBOL programs favored a

statement called the *GO TO* statement. Effusive use of the *GO TO* statement makes for a program where the logic weaves in and out and back and forth and is very time-consuming to trace. Most programmers only have ten fingers, five on each hand. As we read the statements in the code, following the logic, we use our fingers to mark our place when the code uses a *GO TO* construct to branch from one page to another. When we are out of fingers, we lose our place. Many COBOL programs written in the 1970s exceeded this ten-finger limit. This characteristic of the code drives analysis costs through the roof. Although there are tools to perform some of the analysis, the fields that were missed must be reviewed by a programmer. (I am halfway kidding about the fingers—some of us were promoted to paper clips. You can comprehend the culture. . . .)

- Yesterday's programs had funny ways of using date fields. The date 9/9/99 is often used as the date for an *infinity* of sorts. If you didn't want a tape erased for a long, long time, you could use 9/9/99. What will happen on September 9, 1999, which is about a year away? Some very critical business tapes could plummet into oblivion! There are dozens of these values, such as 00, 98, and others, which all mean something special in these old programs and will bingo into unwanted actions if not corrected earlier.
- The COBOL programming language was updated at least once in the 1980s. Some departments updated all of their code to reflect the new changes and recompiled all of their programs. Others didn't. Programs that haven't been compiled in years may require further updating beyond y2k date changes.
- The y2k dilemma is not just restricted to old COBOL programs. Other languages that businesses rely upon include FORTRAN, PL/I, Assembler (of which there are dozens of versions), Ada, Jovial, PASCAL, and the more recent C and C++. At least two languages are found in 30 percent of software applications in the United States and Capers Jones has seen as many as twelve languages being used in one single system.¹⁶ Although there are fewer applications running

Assembler and PL/I programs, there are also fewer people who still know these languages. A dozen years ago, it was hard to find Assembler programmers. Companies still running this code should start looking now for qualified resources to update their programs (see figure 3.11).

FIGURE 3.11: PROGRAMMING LANGUAGES AND APPLICATIONS IN USE TODAY¹⁷



- In legacy systems, the source code is only one piece of the puzzle that must be changed. In an online system, the screens must be changed, and in a batch system the *job control language* must be changed. Database components must also be changed. File definitions must be changed. Even data entry forms must be changed. There are literally hundreds of items in a system that must be corrected for the y2k problem.

As you can see, programming exists in a very detail-oriented world. Once I left out an asterisk in column seven on one line in a COBOL program. This little bitty oops brought down two divisions that could not place orders for thirty-six hours. One little bitty asterisk wreaked havoc on me as well as the Fortune 500 company I worked for.

- As changes to this old code are made, additional errors in logic could be created. One rule of thumb cited in a y2k paper on the Internet states that if a programmer studies one hundred lines of code and makes one change, there will be a one in six chance of introducing another error in the code.¹⁸
- Add the above considerations to one more—the fact that you have a lively, active business, with more going on than just the y2k project. Since the y2k project will touch just about every program in the house, it will overlap with other projects

that are also in progress. If a program is being changed for both y2k compliance and for another mission-critical project, the changes must be coordinated between the two projects. This is called version control.

The good news here is that there are numerous tools to aid COBOL programmers in the changes they must make. These tools attempt to automate various tasks on various phases of the y2k project. Chapter 20 covers this subject, “Tools,” in more detail.

Through the years, businesses have solved various problems with technology and have ended up with multiple hardware and software platforms. Although COBOL on an IBM mainframe was ubiquitous in the 1980s, it was only one of many platforms a company used. In the two large shops that I’ve worked in, I’ve seen AS400s, Unix, Unisys, IBM 40XX, IBM 30XX, local area networks (LANs), wide area networks (WANs), and PCs, all of which must be analyzed.

Although the y2k dilemma is primarily a mainframe problem, other platforms such as client/server will need attention as well. A Morgan Stanley survey, summarized in table 3.1, shows that there will be a large amount of work converting PC-based client/server systems.¹⁹

TABLE 3.1

<i><u>Platforms/Applications to Be Converted</u></i>	<i><u>%</u></i>
Mainframes	51
PC workstation-based client server	25
Minicomputers	18
Other	6

One shop that reengineered a COBOL application to a client server environment carried its y2k problem into the new system. The client/server application was converted to C++, in which four-digit years were used, and Visual Basic, where the two-digit years were continued. The programs written in C++ are OK, but the Visual Basic programs require modification.

For each platform the evils of the y2k dilemma must be assessed. This includes the hardware, operating system, utilities, security programs, scheduling programs, library programs, compilers, and applications software.



SYSTEMS TO STUMBLE ON

Hardware and software today mean much more than PCs and mainframes. We can easily overlook the large volume and variety of embedded systems. We probably do not think about the chips in our sprinkler systems or our elevators. But they are there. The systems some companies might forget to check for y2k compliance include business equipment, security systems, other building systems, and industry-specific systems.

Business Equipment

Almost every company has a phone system. These systems should be checked to make sure they are y2k compliant. If the system is not corrected for the y2k problem, it could put the wrong message on the wrong day. For a small business reliant on phone orders, that's a big deal. Billing details could become erroneous. Date-stamped messages in voice mail could also be incorrectly recorded. Even if a company corrected all of its equipment, the telecommunications companies' networking equipment could fail.

If a noncompliant system is used in a city's 911 operation, any malfunction can become critical.²⁰ Federal Communications Commission (FCC) Commissioner Michael Powell was quoted in *USA Today* as saying "the vast majority of police and fire equipment is not year 2000 compliant."²¹

Lucent, Nortel, and Mitel, large PBX (private branch exchange) companies, have published their y2k plans and notified companies of their products' compliance or noncompliance. Most expect to ship upgrades by late 1998.²² Both Nortel and Mitel have excellent y2k information on their Web pages. A quick search for (y2k) yielded a list of products with model numbers and version numbers, whether they were compliant or not, and when an upgrade would be ready.

To find out whether your phone system is compliant, call your vendor or look on its Internet page. If your system requires a fix, install it as early as possible and test it to make sure it works with next-century dates. To find out how the telephone companies that you rely on for service are doing on y2k compliance, call them. For legal reasons, you'll probably hear, "We're on schedule," from all of them whether they are or not. Your y2k contingency plan should include a strategy for coping with any interruptions in service, whether from your own equipment or from the telecommunications companies' networks.

Other business equipment that should be checked includes copiers, faxes, and postage machines. Most companies are categorizing these systems as noncritical and low priority. They will probably not receive any attention unless they break. If there is time to address these systems, a simple test of entering the date or a phone call to the vendor should provide compliance information for the equipment in question.

Security Systems

If you work in a building that requires card-entry during off-hours, the card-entry system should be checked for y2k compliance. Worst case is your card will think it's either expired or it's Sunday when it's really Monday and will lock you out of the building. If your business is in a high crime area, try not to use such a card in the middle of the night for the first time after the century change.

Other Building Systems

Other building systems could simply cause nuisances and should be checked. These include heating and air-conditioning systems, sprinkler systems, and anything with automatic sensors, such as lights. Electronic time clocks and parking lot gates are also suspect.²³ If you run a nursing home or hospital, these systems must be analyzed. The solution is to contact the vendor and find out whether there is any corrective action to be taken on the system prior to 2000.

Elevators have gotten a lot of publicity. Supposedly, they will place themselves out of service at midnight 2000 if they are not fixed because the software chip will think they haven't had their routine maintenance in 100 years. I could not reach anyone for verification on this issue, but it will not hurt to get a letter from Otis or your elevator vendor that says your elevators will work fine after 2000.

What will my bank vault system do after 2000? This is a question I would surely ask if I worked in a bank. A letter from the company that manufactured the vault is an essential piece of research in a y2k project.

CHAPTER 4:

INSIDE AND OUTSIDE THE ORGANIZATION

Manufacturers, especially large ones, occasionally develop problems meeting production schedules when one of their parts suppliers experiences a strike or fails to receive raw materials. Headlines warn that companies sometimes temporarily shut down plants until a parts supplier catches up. As an example, here is one about General Motors that ran in *The Wall Street Journal* on July 25, 1997: “General Motors is expected to have closed six assembly plants by this morning as a result of parts shortages from a strike at a transmission plant.”¹

The same thing could happen with companies in a supply chain that do not bring their systems into the twenty-first century. These companies are connected electronically, often with electronic data interchange (EDI) systems. The chain can be many links deep for some companies, as suppliers are connected to other suppliers, who are in turn connected to manufacturers.

Sears is a premier example. Not only does it have a large internal challenge with all of its Assembler code, says Capers Jones in a *Computerworld* article, but it has a huge number of trading partners, consisting of suppliers, suppliers’ suppliers, and suppliers’ suppliers’ suppliers.² American Airlines’ Sabre reservation system is another patchwork connecting over 200 hotels, 60 car rental companies, and 400 airlines with travel agents and their customers. A date failure in any one system can disastrously affect the others.³

The more dependent the supplier is on one customer, such as a parts plant for General Motors, the more the large customer can influence the supplier as to the importance of becoming y2k compliant. In an industry such as financial services, where many businesses buy and sell from each other in a tangled web, a business will find it difficult to track down everyone it connects with.⁴

Companies are most concerned with the correct interpretation of shipping dates. In timesensitive markets such as a movie release, Sears cannot afford to have the movie’s promotional products ship off-schedule because of a computer glitch. Boeing has similar constraints. Even one key parts supplier can throw off the entire manufacturing schedule at Boeing.⁵

To gain the offensive, Boeing's Wichita plant mailed 4,000 questionnaires in 1995, but only received 189 responses. Merrill Lynch is sending out test scripts to its 800 business partners and plans to schedule testing with key partners in 1999. Sears has set up a council that meets a few times a year to educate its thousands of suppliers. It will send its auditors to visit a few key accounts. It is also considering setting up a certification program that will be required before a company can do business with Sears. All of these vendors were considering severing relationships with companies that do not become compliant within the necessary time frame.⁶

As part of the analysis process, companies must identify all of the electronic connections it has with external entities. A notification letter should be sent to each business partner asking when its systems will be compliant. Responses to these letters should be acted on diligently. Larger trading partners should receive a personal visit from a y2k-team member.

Contingency plans, should a supplier not reach compliance in time, should be created. A list of alternate suppliers should be prepared. Decide ahead of time whether you are willing to dump a current supplier that did not reach compliance in favor of an alternate supplier who is compliant. If the issue is a large one for your company, dedicated staff should be assigned to this portion of the project.

Project dependencies will be another area to watch closely as y2k projects move into testing phases. Banks are reliant upon telecommunications companies to send electronic funds transfers between branches. The Federal Reserve Board requires that banks be compliant by December 31, 1998; however Bell Atlantic's network, for one, wasn't expected to be compliant until mid-1999. As of mid-1998, the banks, the Federal Reserve, and the telcos were working on the schedule.⁷

Two recent events demonstrate how interconnected we are. The UPS strike affected thousands of businesses, and the satellite that spun out of control affected a host of electronic devices from pagers to television broadcasts. Are these events a sampling of what could happen on January 1, 2000?

CHAPTER 5:

2000 IS A LEAP YEAR

Every fourth year is a leap year. During our lifetimes in the late twentieth and early twenty-first centuries, that's all we really have to know. But the actual rules are more complicated than that.

Today's system of calculating leap years was created in 1582 to correct for the fact that the Julian calendar year was 11 minutes and 14 seconds longer than the tropical (solar) year. Over the centuries, this meant that the spring season was actually arriving earlier and earlier than March 21, the calendar arrival of spring. The calculation of leap years could make the calendar year more closely approximate the tropical year in length.¹

Pope Gregory XIII, advised by a Neapolitan physician and a German astronomer, decreed the following leap year rules:

1. Every year divisible by four is a leap year (1988, 1992, 1996), except
2. Every centesimal year (a year that ends in "00") is not a leap year (1700, 1800, 1900), except
3. Every centesimal year that is divisible by 400 is a leap year (1600, 2000, 2400).

With this correction, a Gregorian year is, on average, 365 days, 5 hours, 49 minutes, and 12 seconds, which is 25 seconds off from a tropical year. Now, every 3,400 years, spring might come one day early.²

Another one-time correction was made to advance the calendar from October 5, 1582 to October 15, 1582. Good thing they did that before computers were invented.³

What does all this mean for the y2k problem? In addition to checking programs for the correct presentation of the year, any programs with date logic must be checked to see if the leap year rules have been correctly followed.

Some programmers became creative with the leap year logic that they wrote in programs. Sometimes the dates of 88, 92, and 96 were handled specifically as leap years, perhaps in hopes that the program wouldn't last any longer than 96. Other programmers wrote in the first two rules above,

but did not know about the third rule. Both of these programming approaches will yield incorrect results in 2000.

Even experts have been confused on the leap year issue. *Computerworld*, which has done a great job of covering the y2k issue for the information systems profession, ran a letter from a reader who cited the *Encyclopaedia Britannica*, and stated that February 29, 2000 is not a valid date.⁴ A week later, they ran the correction, after being deluged with letters from readers who were in the know.⁵

To set the record straight, 2000 is a leap year. There will be a February 29, 2000.

CHAPTER 6:

AND NOW FOR THE SILVER BULLET

There is no silver bullet for the year 2000. As you have already learned, the problem is too pervasive and too complex. Certainly there are tools that will reduce the length of a y2k project, maybe by as much as 30 percent. But there will be no silver bullet. Unless of course, you are in the media. Here are two alleged silver bullet stories as illustration.

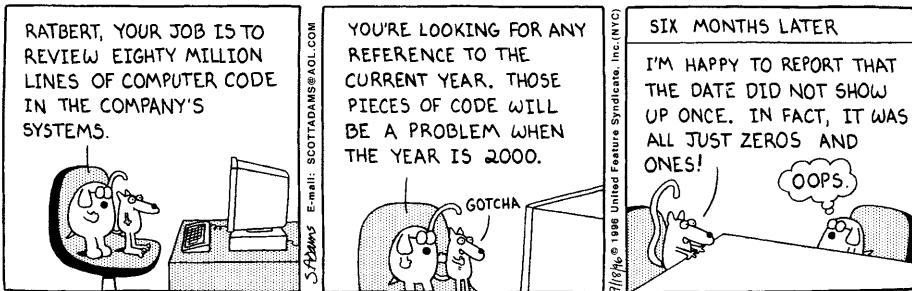
1. *Program Offers Quick Fix to Year 2000 Glitch*, published in *The Dallas Morning News*. Sounds promising, huh? The article describes a Computer Associates International, Inc. tool called CA Fix/2000 that will convert one million lines of COBOL code in less than fifteen days.¹ Even if the product delivered on its promise, the coding phase is only 10 percent of a y2k project. What about testing? You must still perform the time-draining task of testing the one million converted lines of code. What about all the non-COBOL code? PCs? Hardware? Embedded firmware? This may save time on a project, but it is no silver bullet.
2. *Bob Bemer Aims 'Silver-Plated Bullet' At Year 2000 Problem*, published in *The Wall Street Journal*. Seventy-seven-year-old Bob Bemer, a former IBM executive and computer programmer from way back, knows PCs at their very basic machine language level and says he can apply a patch to dates at that level to fix the problem. The article did not say which machines Bob could write patches for. But, again, this solution only aids the coding phase of the project. This patch would have to be tested, which is normally a chunky 50 percent of the project. As an old programmer myself, this solution scares me. How could an auditor audit a patch to the object code of the computer? Bob has two partners, Millenium Consulting Group and Systems Source Inc., both in Dallas. He was scheduled to finish the code during the summer of 1997.²

Expect to see many more alleged silver bullets in coming months reported by the media and by companies that sell the products.

SECTION

I

YEAR 2000 ISSUES



Dilbert reprinted by permission of United Features Syndicate, Inc.

This section will present the issues surrounding the y2k problem. I'll examine several perspectives:

- Accounting
- Legal
- Insurance
- Human resources
- Project management
- Enterprise

I'll discuss the issues related to each of these perspectives in the chapters that follow. But I'll start with the most perplexing issue of all: the issue of awareness and the behavior that follows—or sometimes doesn't follow—as we will see.

CHAPTER 7:

AWARENESS AND THE ANTICS THAT FOLLOW

Some of the executive behavior that I have witnessed regarding the y2k issue seems illogical, confusing, and to some degree, suicidal. Often, when an executive discovers that his or her company has a y2k problem, nothing happens!

Certainly it is shocking to discover that this problem is going to cost a company money and there is no payback except to stay in business. Certainly the word “nightmare” arises, and there is a faint hope that the whole thing will disappear if ignored. The thought that this is all a scheme by systems people (which they have never understood anyway), consultants, and vendors to get rich also justifies inaction.

Whatever executives are thinking, one thing is clear. Many have not yet done what it will take to keep their businesses running past the millennium.

In the first part of this chapter, I will raise more questions than answers. Many of my ponderings are philosophical, and you are urged to draw your own conclusions. In the last part of the chapter, I will present a checklist designed to inspire y2k action.

When did executives first become aware of the y2k problem? Table 7.1, created by International Data Group, identifies when chief executive officers (CEOs), chief financial officers (CFOs), and chief information officers (CIOs) first learned about the problem.¹

TABLE 7.1

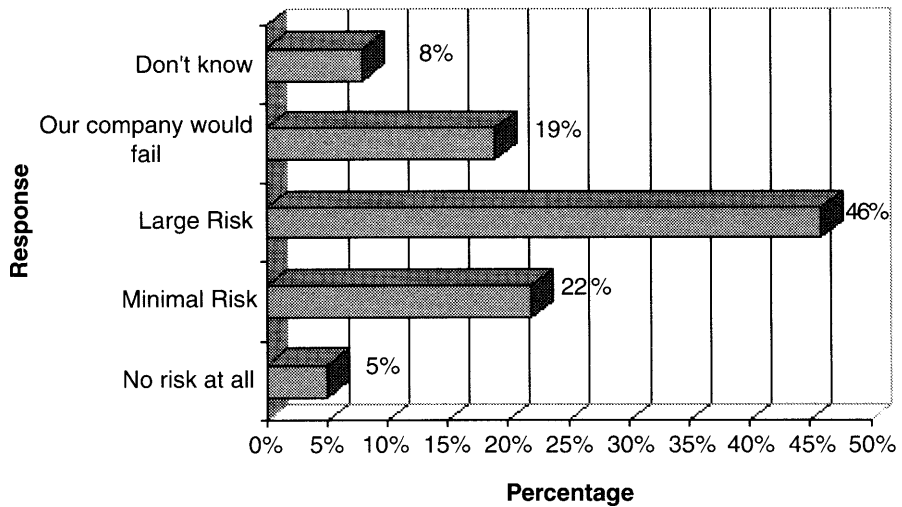
<u>%</u>	<u>CEOs</u>	<u>CFOs</u>	<u>CIOs</u>
Before 1990	8.3	3.1	37
1990	4.2	4.1	8
1991	3.1	2.1	8
1992	3.1	7.2	7.2
1993	7.3	9.3	10.8
1994	30.2	37.1	13.1
1995	32.3	28.9	14.1
1996	11.5	8.2	1.8

Did CIOs fail to alert the rest of the executive team? Did they try? Did anyone listen? Were they laughed out of the executive suite? Did they get

sidetracked on budget issues? Did they think it was not that big of a problem?

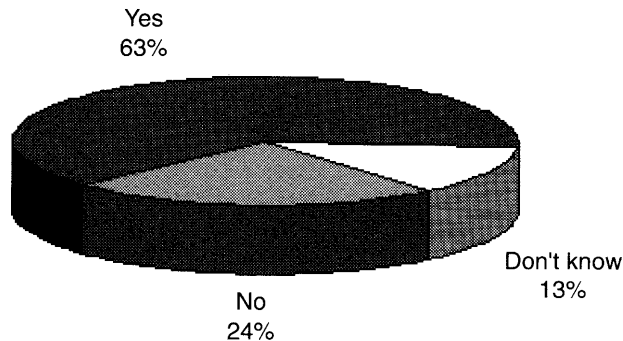
A y2k survey asked the following question to over 1,000 respondents from companies around the globe: “Do you believe the y2k problem poses a risk to your organization, assuming you make NO changes to your computer system?” The answers are shown in figure 7.1.

FIGURE 7.1: RISK OF THE Y2k PROBLEM TO THE ORGANIZATION



The next question on the list was: “Do you believe your company’s executive management is aware of the real risks involved in this project?”² The answers to this question are shown in figure 7.2.

FIGURE 7.2: EXECUTIVE AWARENESS OF Y2k PROBLEM



A Morgan Stanley survey of members in its CIO council reported in 1996 on the following questions:

1. How would you rate the y2k issue?

<i><u>Response</u></i>	<i><u>%</u></i>
A very significant and risky project	38
Just another project	8
Insignificant, a lot of hype	0
Significant, but already under control for us	54

Notice how no one responding to this survey thinks the y2k problem is a lot of hype.

2. Are your CEO and CFO aware of the issue?³

<i><u>Response</u></i>	<i><u>%</u></i>
A lot	62
A little	38

In early 1997, according to a Cap Gemini survey, only 18 percent of Fortune 500 companies had a detailed conversion plan in place for their y2k project. And only 40 percent of respondents had tried to assess the impact of the y2k problem on their systems.⁴

In 1998, there was a dramatic improvement in the numbers, but there was still a long way to go. As of April 1998, companies that said they had a y2k strategy jumped to 60 percent from 20 percent just four months earlier. The number of companies with detailed y2k plans in April 1998 was 68 percent, up from 33 percent in December 1997.⁵

Some 40 percent of small businesses consciously plan to ignore the y2k dilemma, according to a survey conducted by Wells Fargo and the National Federation of Independent Business.⁶

Based on these survey results, a Gartner Group statistic looks plausible; that half of all businesses won't be ready for the year 2000 and one-third of mission-critical systems won't be converted.

Michael Budzich of Gemko Information Group, Inc. in Buffalo, New York categorizes businesses into three groups:

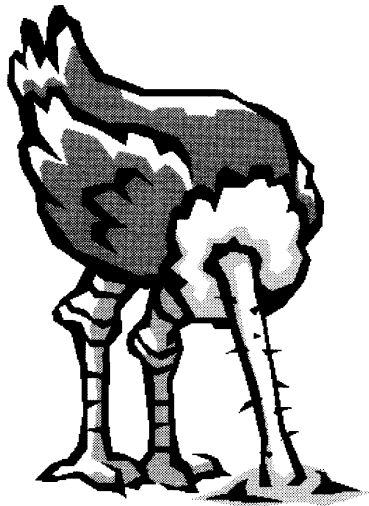
- 1. Businesses that are totally unaware of the problem**
- 2. Businesses that are actively addressing the problem**

3. Businesses that are in denial: the ones that are aware of the problem but are not taking any action to circumvent the problem

Each type of business requires a different approach from Gemko, which sells y2k consulting services and is a consulting arm of Gaines, Metzler, Kriner & Co., LLP.

For purposes of this chapter, I'll focus on the problem spot of executives in denial, or as we say in East Texas, the ones who are frozen like deer in front of bright headlights (see figure 7.3).

FIGURE 7.3: PROGRAMMERS' UNIVERSAL SYMBOL OF EXECUTIVE DENIAL ON A Y2K PROJECT



IN DENIAL

Business executives who understand the problem but do nothing about it require special handling. Some of these executives are technology illiterate. Others are not willing to admit that their business is so dependent upon technology. Luckily, it is not necessary to completely understand how an executive's mind works or why he or she is stuck in denial. It is necessary to perform as many of the following actions as it takes to stir the executive into action on this issue. These same steps will also be useful if you find yourself crying "Show me the money" on a y2k project that is underfunded or still unfunded.

1. Find a system that will definitely fail if it isn't fixed. Demonstrate the failure as graphically as possible. Include how the failure will affect the business' revenues and profits. Companies that started early on their y2k problems generally were alerted to the problem when a live production system failed. Simulating this trigger event will be the next best thing.
2. Repeat step one a few more times until you see the executive begin to move through the classic stages of anger, denial, acceptance and grief. Then continue with step three if necessary.
3. Document what competitors are spending to fix their y2k problems and present this to the executive.
4. Gain buy-in from middle and upper management levels through a series of one-on-one meetings that demonstrate systems that will fail in their functional area and the impact on the business. Then gang up on the executive with your newly converted cohorts.
5. Clip newspaper articles from mainstream newspapers, magazines, and trade press that quote experts about the year 2000.
6. Hire a consultant to run interference and educate the executive.
7. Warn the executive about potential lawsuits and liability if due diligence is not followed.⁷
8. Ensure that the executive understands the pervasiveness and the magnitude of the problem.
9. Gain the executive's commitment to be the executive sponsor of the project. Make sure everyone in the company is notified of this role.
10. Offer preliminary budget estimates, stating that these will change as each project phase is completed.
11. Educate the executive on items that will need to be purchased, such as hardware, software, tools, and training.
12. Present an estimate of staff requirements and whether these estimates can be met with existing staff or must be filled with additional hires or other outside resources. Include

time needed from user department staff to help with identification, design and testing phases.

As a CPA, you can be instrumental in alerting your clients to their y2k issues. There is very little time left for organizations to address the y2k dilemma.

CHAPTER 8:

ACCOUNTING ISSUES

The American Institute of Certified Public Accountants (AICPA) has issued a publication, *The Year 2000 Issue: Current Accounting and Auditing Guidance*,¹ that summarizes the various applicable accounting, disclosure, and auditing standards; describes the responsibilities of various parties; and clarifies the auditor's role. It also provides guidance to auditors on communicating with client management and describes disclosure considerations and practice management matters relating to the year 2000. In my opinion, this paper is a must-read for all accountants. It is available for downloading at the AICPA web site at <http://www.aicpa.org>.

After meeting in July 1996, the Emerging Issues Task Force (EITF) of the Financial Accounting Standards Board (FASB) released Issue Number 96-14, *Accounting for the Costs Associated with Modifying Computer Software for the Year 2000*. This issue states that costs associated with modifying internal-use software to correct for the y2k problems should be charged to expense as incurred.

The task force did not address the treatment of purchases of new hardware or software to replace older noncompliant systems. The common practice is to capitalize these acquisitions and amortize them over their estimated useful life. The AICPA Accounting Standards Executive Committee's Statement of Position 98-2 (SOP 98-2), *Accounting for the Costs of Computer Software Developed or Obtained for Internal Use*, provides guidance in this area.

Software vendors will still follow FASB's Statement of Accounting Standards Number 86, *Accounting for the Costs of Computer Software to Be Sold, Leased, or Otherwise Marketed*.

From a financial reporting standpoint, the advantage lies in replacing systems instead of fixing them, even if the price tag of the new system is higher. A company's financial statements will look better, at least in the first few years of the system, as the replacement costs are spread over a longer period of time than the repair costs. For example, consider a company that has a choice between a \$10 million replacement cost or an \$8 million dollar repair bill. The impact to the bottom line is as follows:

<u>Impact on Profits (in \$000,000)</u>	<u>1998</u>	<u>1999</u>	<u>2000</u>	<u>2001</u>	<u>2002</u>
Cost to replace \$10MM	(\$2)	(\$2)	(\$2)	(\$2)	(\$2)
Cost to fix \$8M	(\$4)	(\$4)			

Obviously, other factors will play a role in deciding whether to repair or replace, but the impact on the financial statement is an important consideration.

Should a company's y2k plans be disclosed in financial statements or reviewed as part of its audit? Currently, there is nothing in the accounting literature that specifically addresses the reporting of future costs of modifying internal-use software. The SEC in Staff Legal Bulletin Number 5 requires a discussion of forward-looking data in Management's Discussion and Analysis of Financial Condition and Results of Operations (MD&A) on Forms 10-Q and 10-K. A discussion of y2k costs is required if those costs are significant.² In June 1998, the *USA Today* reported that the SEC may toughen its guidelines once again because firms were not disclosing enough information about how the y2k problem was affecting them.³ The Audit Issues Task Force (AITF) of the Auditing Standards Board (ASB) has issued several Interpretations of the auditing standards. These Interpretations discuss how the y2k problem may affect an audit of financial statements and address the following areas: the auditor's responsibility, audit planning considerations, communication of matters identified during the audit, and the auditor's consideration of an entity's ability to continue as a going concern. Y2k interpretation of Statement on Auditing Standards (SAS) No. 70, *Reports on the Processing of Transactions by Services Organizations*, and Statement on Standards of Attestation Engagements (SSAE) No. 8, *Management's Discussion and Analysis*, provide guidance on engagements performed pursuant to those standards. These interpretations are a must-read for auditing of financial statements.

The Internal Revenue Service (IRS) is struggling to complete its y2k remediation project. If some systems are not corrected in time, some of the IRS's functions may become impaired. These include investigations, compliance, audits, and customer service. Worst case, collections and refunds could be affected.⁴ Some of this could affect CPAs and their clients.

**CAVEAT**

This brief overview of existing guidance related to the y2k dilemma is for information only. It has no official or authoritative status.

CHAPTER 9:

LEGAL ISSUES

A few lawyers are drooling. A few others are saying the problem is not a problem.

To whom should you listen? In this chapter, we'll present what is contained in the general body of literature and press on y2k legal issues. No legal opinion or advice will be presented here. Please obtain private counsel if your company has any legal issues concerning the y2k dilemma.

You might have already seen some sensational headlines about the litigation that could result from the y2k dilemma. Jeff Jinnett of the law firm LeBoeuf, Lamb, Greene, and McRae stated that litigation costs could run as high as \$1 trillion in the United States. He said the y2k dilemma will be much larger than all of the other liabilities including asbestos, tort, and pollution combined, which have totaled \$300 billion. Managers who neglect the problem will be sued by shareholders, software and hardware companies will face product liability suits, and y2k consultants could encounter malpractice charges.

Steven Hock, head of the y2k team at Thelen, Marrin, Johnson, & Bridges in San Francisco, arrives at that same \$1 trillion estimate, and almost the identical quote.¹ Is this a conspiracy or what?

Capers Jones goes further. He estimates legal fees at \$2 trillion and damages at \$100 billion between 1997 and 2005. A Fortune 500 company, major bank, or insurance company should expect to shell out \$750,000 per year for those eight years.²

The following six groups could suffer most and seek litigation:

1. Clients whose finances have suffered
2. Shareholders whose companies missed the deadline
3. Estates of individuals who died, or individuals who were injured as a direct result of a y2k problem (See the sidebar entitled "Software that Kills.")
4. Customers with class actions against software firms
5. Customers of y2k consultants who missed the deadline
6. Customers of hardware companies producing defective microcode³

We'll discuss the groups with the most liability:

- Directors and officers
- Hardware and software vendors and consultants
- Y2k consultants and vendors
- Entities that certify y2k compliance⁴



Software That Kills

Errors in software can range from causing a nuisance to causing injury or death. Since 1986, software bugs in medical devices have caused death or injury in twenty-four cases. Some of the Food and Drug Administration (FDA) product recalls include the following:

- A monitor that failed to sound when a heartbeat stopped
- An intravenous pump that pumped air into the patient when it ran dry of medicine
- A respirator that sent breaths too fast for a patient
- A database of patients, where the medical history became mixed up between patients.⁵

Software bugs have been blamed in the following accidents and near misses:

- A U.S. Scud missile crashed into American barracks during the Gulf War. This flaw in the targeting software of the Patriot defense missile system killed twenty-nine.
- A bus drove off a California mountain in 1991, and seven Girl Scouts died. A glitch in the bus' transmission software was blamed.

- A Boeing 747 scraped its tail on the runway, then flew unsteadily because a computer miscalculated its center of gravity.
- Six cancer patients received radiation overdoses in the mid-1980s as a result of a software glitch. Three of the patients died.⁶

Software designers and managers in charge of these types of projects have a challenging opportunity to perfect their quality control processes. Although the preceding instances were not y2k issues, the y2k problem will require software changes in high-risk devices. Any software that poses a risk to the public and requires a fix for the y2k problem should be thoroughly tested.

Superior testing and quality control of software applications will help to prevent losses from the y2k problem and in general.



DIRECTORS AND OFFICERS

Corporate management that fails at solving its company's y2k problems will be exposed to substantial losses. Capers Jones said that the average Fortune 500 company that trips on this issue will see shareholders spending \$500 million on litigation.⁷

Directors and officers must make reasonable business decisions, and they must disclose material information about their y2k efforts. Already, many managers are performing due-diligence steps regarding their y2k efforts. Many have joined user groups, attended conferences and training sessions, consulted with attorneys, and heavily documented the actions they have taken to reach compliance. These preventive actions should aid in their defense even if something goes wrong.

Those managers who do nothing face the most risk.

Companies that face renewal of their directors' and officers' insurance policies may be asked to complete an extensive questionnaire regarding the status of the company's progress toward y2k compliance.⁸



HARDWARE AND SOFTWARE VENDORS AND CONSULTANTS

Vendors that sell hardware and software must determine whether their product is y2k compliant. If it is, a vendor can disclose this fact in a contract with a purchaser. If a product is not compliant, there are a few choices. The vendor should decide whether it will make the product compliant or whether it will make future releases of the product compliant. If previously purchased product is not compliant and requires the purchase of an upgrade to become compliant, the vendor must decide how to communicate this fact to its customers.

Most individuals are recommending that the best course for the vendor is to provide complete disclosure. If current customers must purchase an upgrade in order for their system to work, they must be notified of the price and timing of the upgrade. The vendor must decide how to handle current customers who choose not to upgrade, but are still on a maintenance agreement.⁹

Most software vendors are declaring that their products will be compliant after a certain release number, and that any earlier releases will not be compliant and will not be supported after a certain date. This decision is primarily driven by economics: it is simply too costly to retro-upgrade previous versions of the software. Nevertheless, there may be some interesting litigation in this area from customers who are on older versions of software that are not compliant. These customers may have chosen not to upgrade their systems because of heavy customization, substantial hardware requirements, or other reasons. Is it the duty of the vendor to provide a product that continues working? Is it the duty of the customer to replace an older system after a certain number of years?¹⁰

Consultants who have been hired to select systems for companies could be found liable for selecting systems that are not y2k compliant. Some companies search for a system through a process in which a request for proposal (RFP) is completed and sent to vendors. The RFP contains the specifications for a system. If the RFP did not contain a requirement for y2k compliance, was the consultant negligent? Soon the courts will begin answering these questions. CPAs who have installed noncompliant accounting systems for clients should make a list of affected clients, decide on a strategy to address the issue, and contact their lawyers for help in wording the client communication.

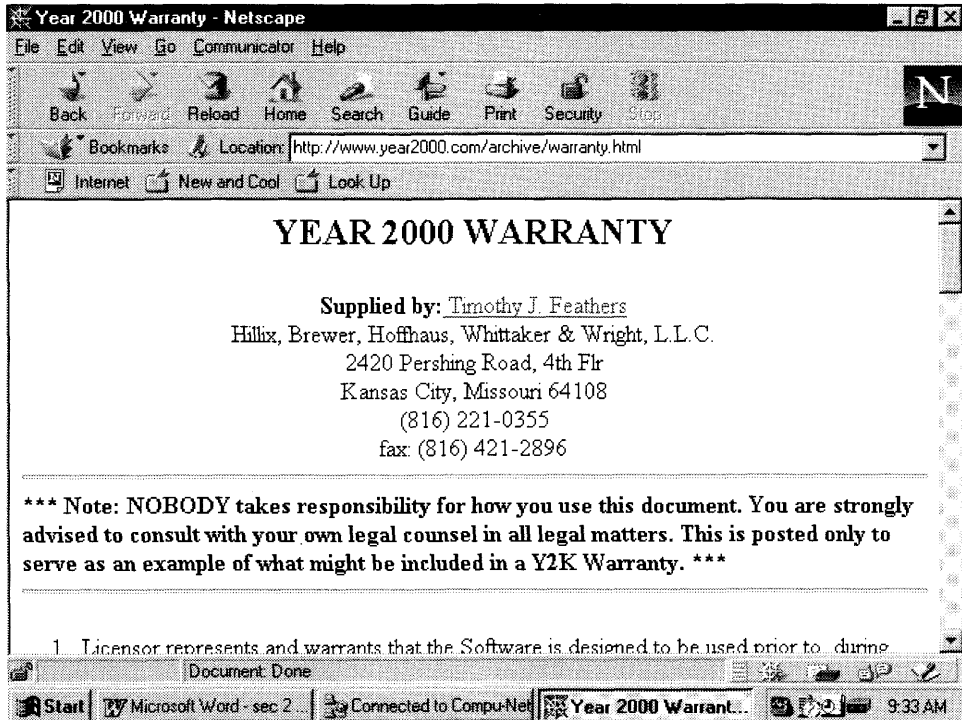
Today's buyers should insist on a warranty in all new contracts. The warranty shall state that the software will work before, during, and after 2000 and that it has y2k capabilities. Terms such as y2k compliance, leap year, and the four-digit year format should be defined. The warranty should include remedies for noncompliance. A sample warranty can be found at Peter de Jager's and the Tenagra Corporation's Web site (<http://www.year2000.com/archive/warranty.html>). (See figure 9.1.)



Y2K CONSULTANTS AND VENDORS

Few companies are outsourcing their y2k solutions, but those that are will be relying on another company's work to literally keep their businesses running. Many issues will arise during the project and into the new century. As consultants change code, a certain number of errors in the work will be likely. As resources become scarce, the qualifications of resources used on the project could affect the error rate.

FIGURE 9.1: Y2K WARRANTY



Complications will arise. What if an unrelated bug is found in the software? Should the vendor fix it? What if the vendor introduces additional bugs when correcting date-related code? If a programmer studies the code and makes one change in one hundred lines of code, there is a one in six chance that a new error will be created.¹¹ This is one reason why testing is so critical on y2k projects. These issues must be spelled out in any contract between a customer and the consultant.



ENTITIES THAT CERTIFY Y2K COMPLIANCE

Some companies are offering a certification that a product is y2k compliant. A company that offers this service should thoroughly examine the code of the product being certified, as well as conduct surprise visits to examine code and documentation in current development.

Buyers should beware of certifications in general. The definition of what's being certified can be misleading. Some companies are certifying a letter from management that the product is compliant without any further independent verification. Others are certifying a business process instead of the product itself.

Vendors advertising certifications may be exposing themselves to a harsher level of warranty should the system fail to live up to its certification.¹²



TO DATE

As of this writing (summer 1998), I know of seven y2k-related lawsuits. The first lawsuit was filed by a Michigan grocery store called Produce Palace International. The store's cash register system, made by Tec-America, the defendant, could not process credit cards that expired in years after 1999. The entire system crashed, long lines developed, groceries melted, and some customers went elsewhere. Because Tec-America wanted \$40,000 to fix the \$150,000 system, Produce Palace International decided to sue instead.

The six other lawsuits are class-action lawsuits against four software companies, of which three are accounting systems providers. Intuit has been named in two lawsuits regarding Quicken versions 5 and 6, which were sold through October 1997. A New York hardware company is seeking \$50 million from SBT Accounting Systems. Macola is being sued. For selling "software you'll never outgrow" and charging for the

y2k-compliant upgrade. Symantec has been named in two lawsuits regarding Norton Antivirus.

The legal fees are mounting. Today, they are mostly in the form of consulting fees paid by clients wondering how they can protect themselves from the onslaught.¹³

CHAPTER 10:

LIFE INSURANCE ON YOUR BUSINESS?

You can take out an insurance policy for y2k coverage on your business. A policy from Minet Group, underwritten by American International Group (AIG), covers lost business revenues, directors' and officers' liability, third-party claims, and losses from problems with electronic trading partners. Coverages from \$10 million to \$200 million are offered. An extensive analysis of the company's y2k plan is conducted by outside consultants to determine whether the company is insurable. A portion of the premiums is refundable in the event that the business continues without disruption.¹

Both Minet and AIG offer separate policies in addition to the joint offering described above.²

J & H Marsh & McLennan, Inc. in New York, offers policies between \$100 million and \$200 million. They are partnering with Ascent Logic, Inc. in San Jose, California, who will perform the required audit, and Leboeuf, Lamb, Greene & McRae, a New York law firm. The audit costs up to \$100,000 plus \$35,000 for licensing software, plus \$30,000 per quarter for an audit update. Premiums were expected to cost in the millions. The standard policy excluded failures from embedded systems in some cases. No policies have been issued to date.³

Most CEOs will probably prefer to self-insure.

One good reason to purchase insurance is the due diligence that it shows shareholders. There is a less expensive option that may gain the same results: certification.



CERTIFICATION PROGRAMS

The Information Technology Association of America (ITAA) in Arlington, Virginia offers a certification on the *process* of y2k compliance. This is not a guarantee that everything will work in 2000. Thirty companies have signed up for the certifications, which cost \$8,200.⁴

Nevertheless, companies that like the idea of certification, which could appease shareholders, should read the fine print and find out what they are really getting.

CHAPTER 11:

HUMAN RESOURCE ISSUES

I'll present three issues that relate to the people on the y2k projects:

1. Finding the people, including a discussion of skills, costs, and availability
2. Choosing the right people for a y2k team
3. Using full-time instead of part-time resources to do the job



FINDING THE PEOPLE

In the search for y2k staff, three issues arise.

1. What skill level does the staff need?
2. What is the market rate for the staff?
3. What is the availability of staff?

What Skill Level Does the Staff Need?

Think about the size of a normal systems project. Then think about the size of a y2k effort. Generally speaking, the larger a systems project is, the more complex it is to manage. It's already difficult to find project leaders with large project experience. Large project experience is defined as at least approximately twenty analyst-years or more. This means that individuals have managed a four-year project with a team of five people, or a two-year project with ten people. With all companies being placed in this condition at once, there will be a shortage of project managers who have successfully led larger efforts. Many companies will have to compromise and settle for individuals with little or no track record.

The ideal project manager will have a solid understanding of the entire enterprise, from a business perspective. The y2k effort is truly a business issue, not just a systems issue. The success of many firms will depend on the success of the y2k effort. A person with knowledge of the business in addition to detailed technical expertise and solid project management experience will make the best project manager. Often, this person will be an employee who has been with the company for a number of years.

There is little time for a new hire to learn the business and run an enterprise-wide project.

An ideal project manager must have excellent communications skills and be able to communicate clearly with both nontechnical managers and executives as well as all technical types, from network administrators to data center operators to programmers to Webmasters. The project manager must be able to develop project plans, assign resources, understand critical path management, and track and report project progress. You might be waiting for me to say this person needs to walk on water, too. Well, almost. As Jim Metzler, a partner in Gaines, Metzler, Kriner & Company, mentions in the *AICPA Solving the Year 2000 Dilemma Videocourse*,^{*} a y2k-project manager with the desired traits is not an easy person to find in today's market.

The Chief Information Officer's (CIO's) skills should be reviewed as well. Many CIOs have never led this large a project in their careers. CIOs also have a dreadful retention rate. A few years ago, it was a revolving door in many organizations. Will you still have the same CIO in 2000? What will the turnover of this position do to your y2k-project schedule? The skill level and retention history of your CIO are factors you must weigh in your y2k efforts.

What Is the Market Rate for the Staff?

What can you expect to pay for y2k staff? According to hiring managers and recruiters in December 1996, the salaries of project managers ranged from \$80,000 to \$100,000. COBOL programmers expected \$60,000 to \$80,000, and y2k consultants were earning \$100,000 to \$150,000.¹ In 1998, when the bulk of the work will begin for most companies, the costs will be driven higher.

Once you have carefully selected the ideal individuals for your y2k team, how can you keep them from being lured away to the next highest salary?

1. Offer a bonus to individuals who remain on the project until completion. Offer a bonus to staffers who bring the project in on time and on budget. Delta Health Systems in Altoona, Pennsylvania, offered their y2k-project team members a bonus of 10 percent of their salary for completing the coding work on time.² The State of Texas' y2k-program office

^{*} Product No. 181800; price to members \$129, non-members \$161. To order call (800) 862-4272.

identified the mission-critical personnel on its y2k projects in various state agencies and offered each of these people a project-end bonus as an incentive for staying through the project.

2. Consider offering benefits and perks, such as allowing individuals to work at home and on flex time. Perks such as new PCs, casual dress, and technology training appeal to systems staff.
3. Make the work environment fun. Y2k work is just plain boring. In a recent survey, on a scale from 1 (boring) to 5 (exciting), y2k work rated a 2,³ so anything that can be done to make the environment more fun and exciting is welcomed by staff members. One Fortune 500 company's y2k steering committee announced a monthly contest in which winners earn restaurant gift certificates, T-shirts, and other prizes. Applicants were required to share their knowledge about their most difficult y2k problem: tricky coding, good solutions, something that would help others, or something that saved money. One applicant wrote up a test problem in which the date, when tested across time zones (where it is yesterday or tomorrow in other countries), was incorrectly assigned 2099 instead of 1999. Seems tricky enough to me!

What Is the Availability of Staff?

As the due date draws near, these highly skilled, expensive staff will become scarce. Systems people are currently stretched worldwide as y2k projects gear up, Internet and electronic commerce projects skyrocket, and Europe braces for the changes that will occur from the adoption of the Euro as the pan-European unit of currency. Capers Jones, y2k expert, estimates that the y2k problem alone will require 700,000 person-years to analyze, correct, and test.⁴ As of early 1998, the United States had 350,000 job vacancies for computer scientists and programmers.⁵ To find people late in the game, companies must become creative. They may resort to—

- Using offshore resources from India, Canada, or Ireland.⁶
- Luring retired programmers back to work.
- Crash-training less skilled workers.

- Tempting employees from other companies.
- Paying large fees to recruiters for finding qualified people.



THE PROJECT TEAM

The business issues that relate to selecting a project team include:

- Who should be included?
- Who should the team report to?

Many businesses are leaving their information systems department in charge of the y2k project. This is a big mistake! The y2k problem affects the entire organization. For example:

- Purchasing policy should be changed across the organization (to buy only y2k compliant products). The company should review key suppliers' abilities to provide an uninterrupted, y2k-compliant supply of products and services.
- Customers may inquire whether your products are y2k compliant. This affects you even if your company is not a hardware or software company. If your company is a bank, utility company, investment firm, electronics store, oil and gas supplier, or automobile manufacturer, consumers are already asking whether your products and services are y2k compliant. This affects sales, marketing, public relations, customer service, and any other department that interacts with customers.
- Legal counsel is required every step of the way, and the legal department must be involved. Risk management issues should be addressed, contracts should be reviewed, and insurance needs should be reevaluated, to name a few issues.
- The accounting department must approve budgets, control expenditures, and record y2k expenses. Internal audit should be included to monitor the changes to the systems and procedures. Credit and collections should ensure that the companies owing large receivables to your company are working on their problem, and the credit policy should be reviewed in light of y2k requirements.

- The facilities or building maintenance group must be involved with the changes required to the physical building and security systems.
- The information systems group will coordinate changes to hardware and software.
- Management of every department must be involved to deal with issues that arise with external entities such as government regulations, key suppliers, and shareholders.⁷
- Disaster recovery managers should be involved to create and implement contingency plans if portions of the project are not completed in time.⁸

Each of these major groups will be represented on the ideal y2k-project team. The most effective team will be made up of individuals who know the business, know their department, have the authority to make decisions for their department, and can make decisions for the good of the business as a whole. A good team will be made up of a manager from each of the functional areas in the business.

As you can probably expect, the project team must report to the top of the organization. The risk of failure of a y2k project is strategic. The CEO must be involved as the executive sponsor of this project.



FULL- OR PART-TIMERS

Many companies have assigned staff part-time duties for the y2k project. For example, a department manager must handle her usual daily work plus spend 20 percent of her time on the y2k project. A programmer working full-time on the y2k project receives additional assignments that cannibalize his y2k-project schedule. Don't fall into this trap!

Can a company really afford to slip those additional projects in and delay the y2k project deadline even further? Does a company think a manager is going to devote 20 percent of his time to y2k activities when daily fires and crises are erupting around him? Both scenarios are a recipe for y2k disaster.

The people on a y2k project must be dedicated, with no other interruptions, and full time, with no other responsibilities. In no other way can staff be held accountable for project slippages.

The only exception to this is a very small company that cannot afford a full time y2k project manager. Small companies can assign the responsibility to an existing staff member, generally an office manager. To avoid project slippage, rock-solid due dates can be agreed upon in advance, and detailed progress reports can be communicated at each major project milestone.

CHAPTER 12:

PROJECT MANAGEMENT ISSUES

The y2k project is basically a traditional systems maintenance project. Some of the organizational and technical issues that normally arise when managing maintenance projects will therefore arise when managing y2k projects. A few of the more challenging ones include the following:

- What is the real deadline?
- Who owns this system? (which translates to, “Who pays for it?”)
- How do we work around a constantly changing company?
- How do we budget for this problem?



THE REAL DEADLINE

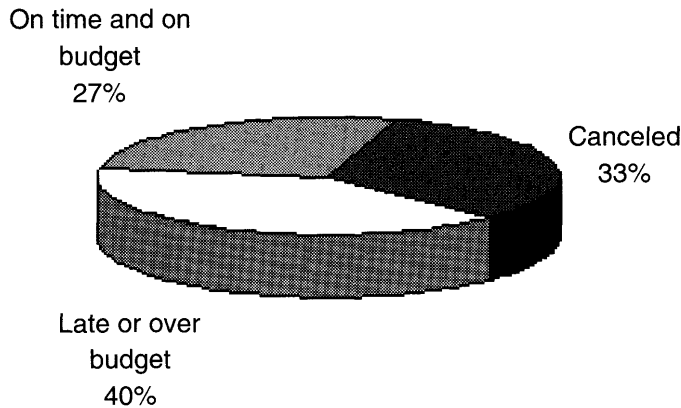
Many companies will wait until the last minute to correct their y2k problems. There will be many excuses, but the bottom line is that most of us are human and we like to procrastinate. The reasons why that will not be smart include the following:

- The cost of everything related to y2k will be driven up greatly as January 1, 2000, approaches.
- Experienced personnel may be committed to other projects and unavailable.
- Products or services to correct the y2k problem may not be able to be manufactured or delivered in time for those who wait.
- There will not be enough time to complete all of the steps necessary to correct the problem if you start too late.
- You could be subjecting yourself and your top management to lawsuits for possible negligence.

A y2k project has a due date that cannot possibly slip. Systems professionals have a dismal record of bringing projects in on time and on budget. In Fortune 100 information systems shops, only 27 percent of projects are completed on time and on budget. Forty percent of systems projects are

late or over budget, and 33 percent are canceled outright.¹ With that record, should you aim for a project end date of December 31, 1999? Definitely not. (see figure 12.1).

FIGURE 12.1: PROJECT STATUS



The y2k compliance work needs to be done much sooner than that. What if account expiration date cycles are three years? What if forecasting systems project five years into the future? What if loan payments are spread over twenty years? Chances are, you've already had y2k problems with systems that track this kind of information. What if your or your clients' fiscal years end in a month other than December? Your y2k problem could begin in that month of 1999 because of the way accounting systems will sort the months ending in 00.

Most companies are aiming for a December 31, 1998 project deadline. That leaves a full year to run the new changes and work out any bugs that were missed. It also leaves 1999 to work out any remaining issues with downstream businesses, such as suppliers, franchisees, member banks, and the like. Since roughly three-quarters of systems projects will be late, the 1998 deadline allows for the slack time that will be needed. However, at this point, a realistic project plan needs to be created. A project manager who simply backs into the project end date without validating the schedule and to appease top management is asking for failure.



THE OWNERSHIP ISSUE

Deep in the bowels of a company's computers, there may be a hidden system, untouched for decades. Maybe one person needs one report from

this system and so it chugs along, producing the one report, and not bothering anyone. Another critical, active system is heavily used. Marketing, sales, and order entry all use the system, as do many other departments.

The questions about who owns these systems are not easily answered unless they've been worked out ahead of time. The ownership questions boil down to the following two issues: Who has responsibility for authorizing changes to the system? Who is going to pay for the changes? These answers are generally worked out in the executive conference room. If your company has ownership issues, allow ample project time to work through the political steps that it takes to get the answers.



A MOVING TARGET

When I was touring across Tibet, we had to cross dangerously high mountain passes, with thin air, freezing temperatures, and ferocious winds. Once, when the driver had reached the very top of a very tall pass at a brain-numbing 17,000 feet, one of the car's tires blew out. Just seconds standing in those winds at that altitude could cause frostbite. Altitude sickness starts at 8,000 feet for people who are not acclimatized to the thin air. My Tibetan guide and my driver said to stay in the car while they got out. I gladly obeyed. A matter of seconds passed before they got back in the car and were off again. No Indy 500 team could have changed a tire faster than they did. Their teamwork was brilliant. It's impossible to change a flat tire with the car running, but the guide and the driver just about accomplished the next best and fastest thing.

Systems shops have to cope with the "change-a-flat-tire-while-the-car-is-running" syndrome daily. Multiple business initiatives are always in progress. The y2k project is one of dozens of projects in progress in the business. Projects overlap in scope. An important program may need to be changed for three different projects at once. The three program versions of changes must be managed. This is accomplished through effective change control management. Since the y2k project is so pervasive, this issue will affect many other projects throughout the life of the y2k project. It's a lot like hitting a moving target.

Other IT projects are being delayed to give the y2k project more resources, more attention, and more focus. Projects that are being delayed include application installations, network upgrades, new PC purchases,

server upgrades, and operating systems and utilities upgrades. Forty-three percent of IT managers in a recent survey said they have delayed other IT projects.²

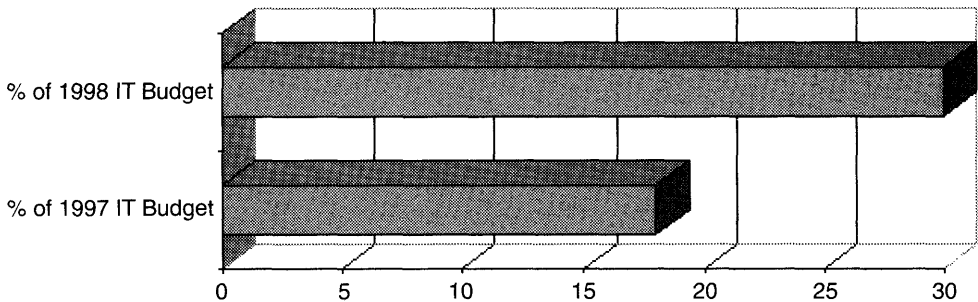


BUDGETING FOR THE Y2k PROJECT

Most companies are taking staff and dollars away from existing projects to fund y2k efforts. In a Cap Gemini poll, companies said they were spending 20 percent or less of their 1997 IT budgets on the y2k project. That percentage is expected to rise to between 20 percent and 40 percent over the next three years, however.³

A Morgan Stanley survey shows roughly the same results. In 1997, the y2k project was 18 percent of IT budgets; in 1998, it is expected to reach 30 percent (see figure 12.2).⁴

FIGURE 12.2: Y2k BUDGET AS A PERCENTAGE OF THE IT BUDGET



In a 1998 poll, 75 percent of Fortune 500 companies surveyed said they will spend between 21 percent and 30 percent of their 1998 IT budget on y2k solutions.⁵ I believe this number is high for Fortune 500 companies, but translates well into what medium-sized and smaller companies will have to pay. Most smaller companies will be forced to hire an expert as few have in-house technical expertise for y2k or even regular IT problems.

How much will it cost to fix your y2k problems? In 1996, Gartner Group provided averages from \$1.10 to \$1.65 per line of code, depending on what modification option you use. Windowing costs \$1.10; date expansion costs \$1.65. (See chapter 18 “Date Modification Options,” for an explanation of the various modification options.) Gartner Group cautions that these figures should only be used as first guesses, and they warn of an error rate of 40 percent.⁶ A Morgan Stanley paper lists line-of-code

estimated costs from \$0.90 to \$1.70.⁷ In 1998, Gartner Group predicted that the cost per line of code to fix the y2k problem would average \$2.00 in 1997, \$3.10 in 1998, \$4.10 in 1999, and \$6.70 in 2000.⁸

Lines of code is a popular, but not a very precise method to break down the size of programs. Not everyone agrees on the length of a line of code in COBOL, for example. Is it the physical line or the logical line of code? In different languages, lines of code are different lengths. Capers Jones argues that function points are a much better measurement than lines of code.⁹

The number of function points in a system is derived from the following attributes:

- Inputs
- Outputs
- Inquiries
- Logical Files
- Interfaces¹⁰

A typical corporate system portfolio contains 500,000 function points. The Microsoft Word program has 5,000 function points. It takes two calendar years to develop an entire system the size of 1,000 function points.¹¹

The estimated cost per function point to fix y2k glitches ranges from \$18 to \$80, depending on the programming language, with an average of \$44. To fix COBOL code, the cost is an estimated \$33 to \$40 per function point. The cost per function point rises as the project size increases.¹²

Whichever method you use, you'll want to work toward converting the ballpark figure into a more accurate budget for your company. After you complete the detailed project plan, a fairly accurate project budget can be created.

The budget approval process is another issue. The y2k project is mandatory for most businesses, but the method in which compliance is performed can vary widely and so can the corresponding costs. An estimate that is accurate enough to distinguish between the major options for systems compliance will aid in making the right decision. An estimate that is too detailed before the final project options have been chosen can delay y2k-project schedules.

A lot of companies are drifting in this step. Precious time is being spent by managers who are having to educate the executives at this late date and cost-justify the project. The reasons for this problem are numerous:

- IT management failed to start the executive awareness process early enough.
- Executives have read that the y2k dilemma is not a problem and now they must be reeducated and reconvinced.
- IT management has several choices to consider about replacing versus fixing versus updating each system in its portfolio, and they started late in their analysis with little staff.
- Executives have ignored warnings about the project, thinking it can't possibly be that much of a problem.

The main thing to remember is to make a fast, informed decision, and move the project to the next phase.

CHAPTER 13:

ENTERPRISE ISSUES

Four issues that affect the enterprise as a whole will be covered in this chapter. First, the risk of business failure as a result of the y2k problem will be discussed. Second, you should be aware of another huge “y2k-sized” project in progress in Europe. Third, I’ll discuss how other countries are doing on their y2k dilemmas. Fourth, the year 2000 doesn’t have to be all negative. I’ll complete this section with a positive spin on the y2k opportunity.



BUSINESS FAILURE OR BANKRUPTCY

Maybe you have seen the headlines (again). Many articles have quoted Gartner Group estimates that half of all businesses will not complete their y2k changes in time. Not all of those late projects will result in bankruptcy, but it is inevitable that a number of firms will fail. One speaker at a meeting sponsored by the Electronic Banking Economics Society predicted that between 1 and 5 percent of firms will become bankrupt as a direct result of y2k costs.¹

Before estimating the percentage of businesses that will fail as a result of y2k issues, Capers Jones likes to break companies into three sizes: Fortune 500, medium-sized, and small companies. He estimates that only 1 percent of Fortune 500 companies will fail. In general, this group is aware of the problem and has already started on repairs.²

Small companies will experience a different set of risks from large companies. Here are four examples.

1. Small companies may not be as aware of the y2k dilemma.
2. Small companies, unlike large companies do not have backups to their systems.
3. If they are unable to service their key customers’ accounts, they will probably lose them. This can have a devastating impact on a small business with only a few key accounts.
4. Small companies have small budgets for such things as y2k problems.³

Nevertheless, companies with less than 100 employees own less software than other companies and generally the magnitude of their problem will be less than that of larger companies. Capers Jones predicts the failure rate of small companies to be an incremental 3 percent.⁴

Companies with between 1,000 and 10,000 employees are at the most risk. They own a lot of software, but do not employ the experts like their Fortune 500 counterparts. They are usually strapped for budget dollars, may not know how to make the necessary contingency plans, and may not get the right talent hired to manage the effort. Capers Jones predicts the failure rate of medium-sized companies to be 5 percent to 7 percent.⁵

Companies in certain industries are at higher risk as well. Banking, health care, insurance, manufacturing, retail, and wholesale are among the high-risk industries. You can guess these high-risk industries; they are the ones that have become software intensive in their processes.⁶

Companies that are already in financial distress could find the y2k dilemma to be their last straw. Companies that are victims of lawsuits may also have to throw in the towel at least temporarily to survive after-2000.⁷



TIME OR MONEY?

Besides the y2k dilemma, there is another large systems project looming in the future for businesses that interact with European entities: the currency conversion to the Euro. As participating countries in Europe move to a common currency, affected businesses must ready their systems for the change. Gartner Group estimates the currency conversion will cost \$100 billion in Europe alone.⁸

The deadline is close at hand. Banks must be able to handle electronic transfers in Euros as early as 1999. Most businesses must convert their currency systems by 2002. Companies' point-of-sale, finance, accounting, payment, and billing systems, among others, will be affected. Desktop systems will be affected too: a new Euro key must be added to all keyboards, and fonts must be updated in all applications.⁹

Many consultants are advising not to fix the two problems at once because the complexity will be too difficult to manage. But they are advocating reusing portions of the earlier project, such as change processes and test environments.¹⁰



GLOBAL MELTDOWN?

However poorly U.S. companies are doing on their y2k initiatives, most foreign countries are faring much worse. From Gartner Group's Compare Scale, here are how a few countries rank on their y2k efforts:¹¹

TABLE 13.1

<i>Time</i>	<i>Countries</i>
In front	United States, and Australia
3 months behind	Canada, United Kingdom, and Israel
6 months behind	South Africa, and Japan
6–8 months behind	Western Europe (except the United Kingdom)
8–12 months behind	Middle East, Mexico, India, and Eastern Europe
12–14 months behind	Argentina, Venezuela, and Central Africa
18–24 months behind	China, Malaysia, Thailand, the Philippines, South Korea, Taiwan, Brazil, and Russia

Only 8 percent of German companies have a y2k plan.¹² In Nepal where the year is currently 2056, in Thailand, where the year is 2541, and in Taiwan where the year is 1987, it's hard to convince people that their computers really will have a y2k problem.¹³ A spokesman for Russia's Atomic Energy Ministry was quoted in the *Dallas Morning News* saying that they didn't have any y2k problems yet and they'd deal with the y2k problem in the year 2000.¹⁴ Businesses with any foreign ties should take extra steps to find out how prepared their foreign business partners are for the y2k dilemma.



TURN REACTIVE INTO PROACTIVE

Although we have focused on problem areas, the y2k dilemma is a potential windfall for some businesses. It provides opportunities for many businesses.

Businesses that have y2k-compliant products and services should highlight this feature in their marketing and use it as an advantage, especially if competitors are behind. A public relations campaign can give credit to the company that is actively working on y2k compliance. The company can inform and educate customers, employees, and shareholders about its y2k initiatives and can ensure the stakeholders that it has the problem under control.

The bad fortune of some companies can be the good fortune of others. It is estimated that some businesses will go bankrupt from not preparing for the y2k problem. This should present some acquisition opportunities for some firms in the coming year or two.

SECTION

III

YEAR 2000 SOLUTIONS



Dilbert reprinted by permission of United Features Syndicate, Inc.

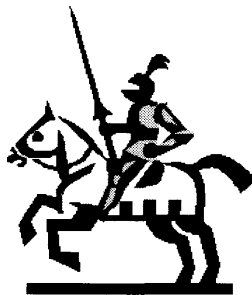
This section will present a step-by-step approach to conquering the y2k dilemma in your firm. The steps we will take include:

1. Systems assessment
2. Project initiation
3. Systems renovation
4. Implementation

Each of these steps will be discussed in detail in the following chapters. Three additional chapters will provide more detailed information about the following alternatives in the assessment stage:

1. Date modification options
2. Y2k consultants
3. Tools

I'll complete the book with a chapter on tips and a look at the future and what we can learn.



THERE ARE MANY DRAGONS TO SLAY IN Y2k WORK.

CHAPTER 14:

PROJECT INITIATION

There's not a lot of difference between starting a y2k project and starting any other large scale systems project. One of the first steps is to assign staff to the leadership positions. An executive sponsor to champion the project must be named. He or she should—

- Approve the project schedule, budget, and staffing before they are submitted to an executive board.
- Clear political issues for team members and maintain the momentum of the project.
- Keep informed as to the status of the project and warn executives of any major issues or changes.

Many companies use a steering committee to act as an approval body for information systems projects. If the company is organized this way, the project should report to the steering committee. This committee will approve the budgets, schedules, and, especially, any changes. This body should be made up of executives from all business units of the company. Meetings should be held monthly or more frequently as the project approaches implementation. Topics to be covered in the meetings usually include project progress, proposed changes, unresolved issues, and any surprises or contingencies.

In a smaller company, the steering committee and executive sponsor might be one person—the owner of the company. In a larger company, the executive sponsor of a y2k project should probably be the chief executive officer (CEO) or chief operating officer (COO). In reality, the chief information officer (CIO) is often given this responsibility.

The project manager is probably the most important position on the project. Most companies prefer an individual who has successfully completed a project of the same size in the past, although this is no guarantee of project success. Communication and leadership skills are more important than technical skills in this position. A project doesn't really get off the ground until a project manager is named.

To start, a project manager must begin planning the project. A very rough schedule and budget, including staffing requirements, should be prepared,

which will be fine-tuned as the project progresses. Once the initial plan is approved, the project staff can be hired or transferred in from other assignments. As personnel are added to the y2k project, a special effort should be made to retain the team members who are critical to the success of the project. This may take the form of project-end bonuses or other perks.

Some companies have successfully set up a y2k-project office. This centralizes the massive amount of tasks that need to be performed relative to the project. The office will not only field technology tasks but will also handle legal issues, media requests, employee education, and vendor requests. It acts as a clearinghouse for all y2k issues in the entire organization. A project office is good for:

- Large companies that are decentralized but wish to centralize the y2k effort
- Companies with a lot of public exposure
- Companies that have substantial y2k problems outside the information systems department

Because a large volume of y2k litigation is expected, project documentation must be meticulously kept to show the steps that the company took to prepare for 2000. Y2k-team members should be trained as to the importance and meaning of their project documentation. Even emails should be carefully worded because they could show up as evidence in a lawsuit.

Other project initiation tasks include the selection of project automation tools, project methodology, and project standards. These steps are routine in most IT projects.

The following five tasks in the project initiation phase deserve extra attention.

1. Company policies should be modified for y2k issues.
2. The company's products and services should be reviewed to ensure that all are y2k-compliant.
3. Awareness letters should be sent to customers and possibly other business partners.
4. The company's strategies should be reviewed in light of the y2k dilemma.

5. Y2k training should be completed.

These tasks are discussed below.



UPDATE YOUR COMPANY POLICIES

You can keep your company's y2k problems from growing by changing your purchasing policy immediately. Put a policy in place that will require y2k compliance of all major purchases that contain computer chips. In the case of large-dollar purchases or mission-critical systems, require the vendor to furnish written documentation that the product is y2k compliant.

A CEO recently held up a high-dollar purchase of a telephone system by bringing up the topic of y2k compliance right before the contract was to be signed. The deal was held up for a few days while the vendor frantically researched the question that it had never been asked before. The CEO signed the deal when the vendor's legal staff furnished a letter that stated that the system was compliant.

Most U.S. government offices adopted a y2k-compliant purchasing policy in 1996. The policies for packaged software and hardware and software services are described in the following sections.

Packaged Software

Before completing a requisition for packaged software, a U.S. government employee must research whether the software version planned for purchase is y2k compliant. This is done by reviewing product literature, Web sites, or other publications.

Hardware and Software Services

The General Services Administration recommends that government agencies that will be purchasing computer hardware and software include the following language in the technical specifications of the purchase proposal.

Computer equipment, software and systems must meet the following requirements:

The software must perform fault-free in the processing of date and date-related data (including, but not limited to, calculating, comparing, and sequencing) by all hardware and software products delivered under this contract/procurement, individually and in combination, upon installation. Fault-free performance includes the manipulation of this

data with dates prior to, through, and beyond January 1, 2000, and shall be transparent to the user.

Hardware and software products, individually and in combination, shall successfully transition into the year 2000 with the correct system date, without human intervention, including leap year calculations. Hardware and software products, individually and in combination, shall also provide correct results when moving forward or backward in time across the year 2000.¹

CPAs can adapt this policy to their own and clients' current purchasing policy. This step is one that should be implemented immediately.

Policies in the credit and collections area should be reviewed in light of the y2k dilemma. Obviously, credit should not be granted to customers who are not on top of their y2k problems. The company should take a fresh look at companies that owe any large receivables and make sure they are making good progress on their y2k project.

Companies with mergers and acquisitions departments should add y2k criteria to mergers and acquisitions being considered.

Insurance coverage should be reviewed in light of y2k issues. Legal advice should be sought early in the project. Y2k certifications should be considered for the company as well as for key suppliers and vendors. The accounting and tax implications of y2k costs should be reviewed in accounting. All other company policies should be reviewed and modified for the y2k dilemma's impact.

Products and Services

The last thing a company wants to do is continue to sell noncompliant products and services and add to the y2k problem. A top priority for a company is to check its products and services and make sure they are compliant. This may mean sending letters to suppliers asking them about parts. If noncompliant services and products have been sold in the past, legal counsel should be consulted and management should create a strategy to handle the customers of these sales.

Customers

In some industries, such as financial services, it will be prudent to send *awareness* letters to customers. This can take the form of a newsletter or letter and should define the y2k problem and briefly outline the solution. You might get a call on January 3, 2000, from a customer saying, "Why

didn't you tell me?'. If this is a possibility, then a letter is a good idea. Moreover, if you would be adversely affected by a customer going bankrupt from the y2k problem, then the letter is a good idea.

Corporate Strategies

Executives should spend some time thinking about the ramifications of the year 2000 on their businesses from a strategic viewpoint. The year 2000 could change strategy by increasing acquisition opportunities, for example. Perhaps it could threaten a foreign division and delay expansion plans. It might refocus marketing plans.



Y2k TRAINING

When I was phoning for information to include in this book, I called the customer service line of a company of which I am a customer. I asked to speak with a person knowledgeable about the y2k problem. First, I had to define the problem to her. Then I was passed to my customer service representative. I had to define the problem for *her*. She had to call me back. At this point, I was really worried about the future of this company. My contacts up to that point surely made the company sound like it didn't have a clue about the year 2000. It wasn't until a week later that my customer service representative gave me the name of a consultant that leads the y2k project for this organization. The consultant was very knowledgeable about what the company was doing and what the initiatives were.

Don't make the same mistake in your company! Customers should not have to educate employees about the y2k dilemma. Although I think it is good that I told two more people about the y2k dilemma, I don't think I want to take on the world single-handedly. Customers should be able to contact a company and find out how the company is doing on this front. Companies should provide their employees with a comprehensive education program so that everyone from the executives to the janitors and the mailroom clerks understands this issue.

Several communications vehicles can be used to get the message out. Depending on the computer culture of your company, a newsletter, intranet posting, training meeting, or a combination of these methods could be best. The notification should include:

- A simple definition of the company's y2k problems
- Examples of what could go wrong
- Who is in charge of the project
- Systems that are being worked on
- What employees need to do to help with the problem
- What policies have changed as a result of the initiatives
- Where to go for further information
- What to do if the employee gets a y2k question from outside the organization

Executives should be trained as well, but on different points. Executives should understand the company's y2k strategy, the project approach, and key risks to the enterprise. They should know the y2k team and its strengths and weaknesses. They should strongly monitor the project schedule and ensure milestones are met. They should keep up with current y2k events and their expected economic impacts.

Specialty y2k training, such as for loan officers in a bank or auditors in a CPA firm, may be required in certain departments. The y2k team may need training too, on topics from project management to best practices on y2k projects.

CHAPTER 15:

SYSTEMS ASSESSMENT

The systems assessment phase involves the following:

1. Taking an inventory to determine the scope of the problem
2. Deciding what action will be taken on each inventory item to make it y2k compliant
3. Setting priorities among systems that require action
4. Completing the detailed project plan and budget
5. Creating a contingency plan



TAKE AN INVENTORY

To start the assessment phase of the project, take an inventory of existing systems. The inventory acts as the basis for determining the scope and extent of the y2k problems you face. The following information about each system should be gathered:

- Name and description of the system.
- Do you plan to have the system in 2000? (If no, list the replacement plans, and stop here.)
- Who owns the system? Which department(s) uses and updates the data?
- How critical is the system to the business?
- When was this system first installed?
- The origin of the system?
 - If the system was purchased from a vendor, list the following:
 1. Company name
 2. Contact name
 3. Phone number
 4. Warranty/maintenance agreement
 5. Any contract information
 - If the system was developed in-house, list the languages used.

- List hardware platform.
- List components of the system.
- Is the system documented or is there anyone in the company that knows the system?

List the particulars.

- If the system contains external interfaces, list the names and addresses of companies using the system.
- When will the system begin failing (December 31, 1999 or sooner?)
- List the degree of data usage (high, medium, low).
- Detail any preliminary knowledge of the system's y2k compliance.

How do you start making your list? Some individuals start with a checklist of categories. Others are literally walking the halls of each of their business locations to arrive at a comprehensive list. That includes listing all components of a system: hardware, software, and firmware. (See appendix A, "Systems Inventory Checklist," for an abbreviated checklist.) One project manager commented in an Internet discussion group, "I am looking at everything that is plugged in." Still others are perusing fixed assets registers. A combination of these methods will ensure the completeness that is necessary for project success.

Once the systems inventory is complete, it will be useful in completing other tasks within the organization including the following:

- Planning for and purchasing software package installations, including operating system upgrades
- Detecting theft
- Providing configuration information to end-user computing
- Tracking moves
- Budgeting and cost management¹

Who should determine how critical the system is to the business? Only the owner of that system. The systems department should have a voice in the ranking, but the owner knows best whether the business will be able to function with or without that particular system.

If a purchased system is several years old, tracking down the vendor could be a problem. What if the vendor has moved, closed up shop, or gone

bankrupt? A trace must be started to see whether any contractual agreements are the responsibility of a new company. Members of your legal department can help to review contract options in these cases.

A spreadsheet or database is useful for gathering the inventory information. Columns and tables can easily be added for more information as needed. As priorities are entered, the data can also be easily sorted in the spreadsheet or database.

Once the inventory is complete, it can be sorted into the following three groups of systems:

1. Systems that you do not plan to have in 2000
2. Systems that are y2k compliant
3. Systems that are not y2k compliant

Systems that you do not plan to have in 2000 require no further action. For systems that are y2k compliant, it is useful to add one more piece of information to the inventory: To what date *is* the system compliant? For example, if a system has been made y2k compliant with the windowing method and the program logic recognizes dates from 1951 to 2050, then the system is compliant until 2050. It will be extremely useful to know when the next y2k problem will occur for this software or hardware. Even if you expect to be retired or dead by then, this will be good information to have recorded. (Obviously, COBOL programmers did not expect their code to live in perpetuity as it has in some shops.)

For systems that are y2k compliant, you must decide whether it is necessary to test the systems for compliance. If the system is mission critical, test runs will validate the claim of compliance or find errors that can be corrected in time. If the system is not mission critical or it can be determined with reliability that the system is y2k compliant, then most companies will probably not spend the time and money to test the system and no further action would be required for these systems.

Before the alternatives can be considered for the systems that are not compliant, notification letters must be sent for purchased systems and systems with external interfaces. The following two sets of letters should be distributed.

1. Software package vendors and firmware vendors must be contacted to determine which release or version of their product is y2k compliant. Once that knowledge is known, upgrading the system can be considered as one of the

alternatives for each software package or embedded system in the company. For example, a bank should send a letter to the company that manufactured the vault containing the safe deposit boxes to determine whether an upgrade is necessary. Sample wording is available on the Internet in numerous places.² Nevertheless, most lawyers advise that they should also review the wording of any y2k correspondence.

2. Suppliers and customers who have systems that interface with your company's systems must be contacted to determine when and how their systems will be changed.

Some companies are having trouble eliciting replies to the letter and are sending as many as three letters to each vendor or business partner.



ANALYZE THE ALTERNATIVES

Now that the inventory is finished, an analysis must be completed that determines the best solution for each system. Not all options will be available for every system in inventory. The nature of each system will determine which option is available for the particular system. The most common options discussed individually in the subsequent sections, are the following.

1. Replace the system (available for either packaged or internally developed software).
2. Upgrade the system (available for only packaged software).
3. Fix the system (available for only internally developed software).
4. Eliminate the system (available for all systems).
5. Wait or do nothing (also called fix-on-failure available for all systems).
6. The solution is outside of the project scope (generally available for systems that are developed by users and that are not maintained by the information systems department, such as spreadsheets).

Each option is discussed in detail, including its advantages and disadvantages, in the following sections.

Replace the System

The replacement option works best if the following conditions are met.

- The existing system has been around for a while, and platform and maintenance costs are already high.
- The existing system does not have the level of functionality that the company needs.
- A replacement system exists in the marketplace that will meet the company's needs or the company has the in-house talent to create a system that will meet its needs.
- The cost of replacing the system is close to or marginally exceeds the cost of repairing the system.
- You were already planning to replace this system, just not as fast, but you can accelerate the schedule.

A company will gain several benefits by replacing a noncompliant system:

- Increased functionality
- Reduced ongoing maintenance costs
- The ability to capitalize the costs over the estimated useful life of the new system

Disadvantages of replacing a system could include:

- Additional hardware may need to be purchased to run the new system.
- Replacement costs may end up being marginally greater than the costs of repairing the system.
- The time and cost of learning a new system must be factored into the project.

Before choosing replacement, a look at the calendar is wise. There must be enough time to select, customize, and install a new system before y2k compliance is required. In most Fortune 500 companies, it is already too late to start a large system installation. The lead times are very long, usually two or more years.

Vendors that offer y2k compliant systems are finding that their sales are rising. SAP, PeopleSoft, and SAS are three examples of companies that have benefited from being on top of the problem before everyone else.

Upgrade the System

If a system is a packaged system, a later version of the package may be y2k compliant. The system can be upgraded with the compliant version instead of being completely replaced.

Many software vendors are currently scrambling to complete their y2k compliant upgrades. Very few are going back and making every version in history compliant. But many companies are still running older versions. This action by the vendors will force some businesses to upgrade, whether they want to or not. Economically, software vendors have all the following advantages if they do *not* make older versions compliant.

- They will sell upgrade packages and increase sales.
- The installed base on older versions will shrink, generally saving the software company the high costs of maintaining these older versions.
- They save the high costs of making older software y2k compliant that have no corresponding sales to absorb the maintenance costs.

Oracle's solution to their y2k problem will force all of their customers to upgrade to the latest version of their software in the next few years. Oracle Applications Version 10.7 is the only version that is y2k compliant. Oracle has not only decided to make only the newest version y2k compliant, but also they recently announced that they will stop supporting older versions altogether in 1998.³

Other vendors are offering the y2k upgrade free if you are a member of their maintenance program. These vendors are selling a lot of maintenance programs because customers wish to avoid costly upgrades.

Fix the System

The fix-the-system option works best when:

- The system is fairly new and has a long life expectancy.
- The system meets most of the functional requirements of the company.
- The system is critical to the business.
- The system is nearly compliant, or the cost of compliance is low.
- There is no replacement on the market for this system.

- The system has good documentation or your staff knows the system.
- The cost of writing or buying a brand new system is prohibitive.

If the choice to fix the system is made, another decision must be made as to how to change the system. There are several date modification strategies to consider. See chapter 18, “Date Modification Options,” for a detailed discussion of these options.

Eliminate the System

Eliminating the system should not be forgotten. Although it’s the cheapest alternative, of course, it can also be very effective. While performing the inventory step of this project, were any orphan systems discovered? That is, were there any systems whose owner was reluctant to come forward and have it inventoried? These systems are prime candidates for being eliminated.

Through the years as additional systems are implemented, the older ones may not quite get turned off in the way they should. The sooner you can pull the plug on redundant systems or systems that no longer generate useful information, the better. If you suspect that you know of a system like this in your organization, have the report distribution stopped on that system and see whether anyone misses their reports! If no one asks about the missing reports, you can probably pull the plug. If only one or two individuals scream, find out what they do with the reports and whether that activity adds value to the organization. A little reengineering could save you from the cost of converting one more system.

Wait or Do Nothing (Commonly Called Phase 2 in Some Systems Departments)

The do nothing alternative may be worth considering in some cases. Many businesses will not have the luxury of making every system in their organization y2k compliant. For these noncritical applications, a wait-and-see approach could be an acceptable alternative. Systems in this category will probably include the following:

- Office machines, such as copiers and faxes
- Systems that generate noncritical management information and do not perform any critical transaction processing

■ Nuisance items like:

- Electronic equipment
- Some building systems such as sprinkler systems
- Some desktop software packages that are not date-critical

Companies that choose this option for some systems should understand the potentially high cost of this option if a system fails. In the case of a system failure, the company could be without the system for some time, until it can be repaired. The cost of repairing a system in a reactive mode could be higher.

Many companies will still be working on their noncritical systems in 2000 and years beyond. Some government agencies' y2k plans reach as far as the year 2014.⁴

Outside of the Project Scope

Some systems will be determined to be outside of the scope of the company's formal y2k project. These systems will be deemed the responsibility of the person who is running the software. These systems typically include the following:

- Macros, databases, and spreadsheets that are created and maintained by the user departments
- Any program files written and maintained by one or two people that do not interface with any other business system

Handling these systems primarily involves distributing a company-wide letter that defines individuals' responsibilities with regard to this type of software. This can be done during the initiation stage of the project as a part of employee education.

If spreadsheets are mission-critical to the company, they should be included in the project scope so that a complete inventory can be taken, remediation can be tracked, and compliance can be audited.



SET PRIORITIES

Now that you have an understanding of what alternatives can be chosen, it is useful to assign a tentative option to each system. You may want to assign two options to each system: a best case and a worst case. There may not be time or budget to do everything best-case.

Rank your systems into three groups:

1. *Systems that are critical to your business.* These are only the ones that move product or services; payroll isn't even in this group for some companies. These are the systems that must be fixed or your business will stop. Decide how much time you have and whether you can do the best-case options or the worst-case options for these systems before the clock strikes midnight.
2. *Systems that are important to your business and would cost significantly if they were shut down for a long period of time, but would not ruin your business.* This may include payroll and other accounting systems. These are systems that you could do without on day one of the new millennium, but you wouldn't want to be without them forever.
3. *Systems that are not worth worrying about until they break.* This could include fax machines, copiers, marketing reporting systems, and spreadsheets. They will be a nuisance if they break, but you can go without them for quite a while or you can easily replace them.

For the kinds of systems that your company depends on, this is a form of triage. To determine how big the y2k project is, you'll want to create detailed estimates and project plans and schedules of each system, starting with the first group. For systems with no clear-cut alternative, analysis should be done to help decide which alternative is best.

One mistake companies are making is getting stuck at this point. You simply do not have six months to kick around alternatives any more. It is important to fast track your decisions and get started as quickly as possible on the systems in group one. One team can get started on the critical systems while another team completes the analysis for the rest of the project. Too many companies have already busted their schedules on this step by overanalyzing their alternatives and waiting for monthly meetings to roll around to push ideas around some more. No one has this luxury any longer.



PROJECT PLAN AND BUDGET

Once the options are chosen for each system in inventory, the project plan and budget can be created. As the project schedule details emerge, the y2k project manager may find it impossible to complete everything by

December 31, 1999. That's OK. If the project leader can complete group one by that date, your company will not fail. Many analysts are estimating that y2k work will extend several years past the millennium change.

Some companies are compressing their schedules by installing temporary fixes to key programs, planning later to install permanent fixes. This is another way to get more systems converted if your company has waited too late for the volume of work it must do.

Another way to compress the schedule is to get a little help from your friends. Included in this section is chapter 19, "Y2k Consulting," including descriptions of what services they offer. Not many companies are choosing to outsource, however. Many reasons suggest it is better to handle this problem internally, including:

- Internal programmers know the business better.
- Many companies have no experience outsourcing and this is not a good time to start.
- Many companies believe they have adequate expertise in house to solve the problem.
- Many companies think the problem is too critical and should not be delegated.

In the case of a small business, it's often better to hire an expert instead of relying on inhouse staff. Most small companies do not have the technical expertise to handle many systems problems, much less the mother of all systems problems: the y2k problem. It's best for small businesses to hire an expert who knows y2k pitfalls and can create an effective y2k plan for the company.

Several hundred vendors are offering automated tools to speed up y2k work. Tools are plentiful in some areas, such as mainframe COBOL platforms, and scarce in others, such as in distributed systems. Chapter 20, "Tools," discusses the tools that are available for y2k projects.



CONTINGENCY PLANS

A contingency plan might be the most important part of any y2k plan, yet 38 percent of Fortune 500 companies have not developed contingency plans.⁵ The contingency plan should be built upon the company's disaster recovery plan and should answer every "what if" question you can come up with relating to the y2k dilemma. What if the project manager quits?

What if the primary supplier goes bankrupt? What if the largest customer cannot pay the bill for a while? What if a mission-critical system is not remediated in time? What if there is no electricity or phone service? What if we miss something?

For each system, a contingency plan should address alternatives with trigger dates that alert team members to implement the contingency. One company bought an accounts payable software package and has installed it in case the custom accounts payable system is not completed in time. The package does not have the bells and whistles the company needs, but will cut checks if necessary.⁶

Staffing contingencies must be in place if key team members leave at a crucial time during the project. Contingency suppliers should be interviewed and preapproved in case primary suppliers fail. Extra paper may even need to be ordered for companies that wish to keep detailed records of customer accounts so that customers' questions can continue to be answered even if systems are unavailable.

Most y2k-team members will be in on alert throughout January 2000, ready to handle unanticipated problems. A crisis team should be developed and trained to provide fast response to problems that arise early in 2000. Executives should be available by pager or cell phone (or better yet, be physically available if telecom systems fail) to make decisions on the spot. Some companies should make a database of staff skill sets so qualified people can be deployed to the crisis areas. A list of key resources with all contact numbers should be printed and distributed to all affected personnel in late 1999. When developing contingency and crisis plans, companies should consider that public transportation and utilities may not be available during crisis times.⁷

In January 2000, security breaches will be more likely than usual because hackers and employees will anticipate that firms will be preoccupied with other things. Y2k-team members should make sure that they do not let their guards down or let system security slacken while a y2k problem is fixed.

All contingency plans should be tested thoroughly.

CHAPTER 16:

SYSTEM RENOVATION

The decisions that were made in the assessment phase are carried out in the system renovation phase. Those decisions were:

1. Fixing systems
2. Replacing systems
3. Upgrading systems
4. Eliminating systems

Each of these four areas will be discussed, and followed by a brief comment about systems with external interfaces. Testing is a large part of renovation and will be covered in the last half of this chapter.



FIXING THE SYSTEM

For those systems that will be fixed, the y2k project will become a fairly large maintenance project. If several systems are involved, it is useful to break the project into smaller, more manageable subprojects, perhaps one for each system. Establishing the priority of the systems to be fixed is important, and key systems should be converted first. If conversion tools will be used, they should be selected and purchased now. Staff should be hired to complete the coding changes.



REPLACING THE SYSTEM

Replacing a system is a complete project by itself. A list of requirements must be determined, a market search must be completed, a pilot is often conducted, legal contracts must be signed. The package must be customized, tested, and installed. Users must be trained.

For each system that will be replaced, a project should be created that is outside the scope of the y2k project itself but scheduled to meet the y2k deadlines. These projects should be managed like regular systems development projects, except to make sure the y2k projects have the highest focus and priority in the company. If a vendor promises a package will be delivered on a particular date in the future, this delivery date

should be closely managed. Vendors that miss their due dates could cause problems to your project dependencies and to your project's critical path. Contingency plans should kick in on "trigger dates" if the vendors miss their dates by a significant period of time.



UPGRADING THE SYSTEM

For systems that can be upgraded, a new software package version must be obtained. (The same problems mentioned above with vendor delivery dates may also occur with upgrades.) A new contract agreement may or may not be required. If it is, a legal representative should review the agreement. The new version is usually installed into a test environment and tested before it is installed into production. Any customization to the package must be made in this phase.



ELIMINATING THE SYSTEM

Perhaps you will be able to discontinue a few systems before 2000. If the system is segregated from other systems and it is clear which modules to delete from the production systems, then there is nothing that needs to be done until the next phase, implementation. If the system is integrated and there is a chance that deleting modules could cause some downstream breakage, then testing should be done in these interfacing systems before pulling the plug.



SYSTEMS WITH INTERFACES

Some systems interface with suppliers' or customers' systems. As the system becomes compliant, the suppliers' and customers' data must become compliant too. Creating two pipelines for entry makes it easier to coordinate changes among everyone involved. One pipeline should be designed for customers and suppliers who are compliant, and the other pipeline should be designed for customers and suppliers who are not compliant. As suppliers and customers convert their systems, they can move from using the noncompliant pipeline to using the compliant pipeline. Eventually, all of the interfaces will be compliant, and the noncompliant pipeline can be unplugged.



TESTING

Testing is expected to take roughly 60 percent of the time of a y2k project. Testing should occur in every system that has changed whether the code has been fixed, a replacement system has been installed, or an existing system has been upgraded. Additionally, a company with mission-critical packages that the vendor has warranted y2k compliant should test those assertions in this phase.

There are many aspects of testing, but it generally starts with the creation of a test plan. The test plan is usually completed at the end of the assessment or analysis phase of a project. It describes what should be tested, who should test it, how it will be tested, what sequence of events should be followed, how the results should be listed, and many other details. A test plan will contain test cases that go into minute detail and will contain such entries as:

- Enter an Order #1 with a December 29, 1999, order date, and a January 5, 2000, shipping date for a chain store customer on the picking screen of the ordering system.
- Change Order #1 to a back order and check the results.
- Change the shipping date of Order #1 to February 29, 2000, and check the results.

These are just a few of hundreds (or thousands) of test scenarios, or test scripts that should be created as part of a test plan. The creation of a test plan is very detailed work. More generally speaking, there are several key areas that a thorough y2k-test plan will cover. A sample of ideas of y2k-specific test suggestions is presented below.

- Check a set of data in the 2001 to 2010 year range for correct processing.
- Check to see the month and day or month and year are never confused.
- Check leap-year days, including the day before and the day after, and check in multiple years that are and are not leap years.
- Check a set of data in 2000 for correct processing.
- Check date calculations of birthdays in both centuries.

- Check date calculations using expiration dates and check the use of expiration dates.
- Check the processing of transactions across the 1999-2000 time frame.
- Check date calculations that span two years, such as order date in 1999 and a shipping date in 2000.
- Check multiyear transactions, such as loans that span thirty years, for correct processing.
- Check transactions that span time zones, in which two different days are happening at once.
- Check the order of transactions sorted by date.
- Check the display of dates. (Make sure it is clear to the user the order of month, day, and year. In 2001, it will be difficult to distinguish year, month, and day since values will be similar. For example, 01-02-03 could be January 2, 2003 or MMDDYY; February 1, 2003 or DDMMYY, European; or February 3, 2001 or YYMMDD, military.) How will you know unless you put the format next to the date values or use the month spelled out instead of the date values?)
- Check dates that programmers use for special meaning, such as September 9, 1999 (9/9/99) to make sure the code doesn't do something drastic like erase all the tapes.

The phases of testing will vary in name from one methodology to another, but some of the more standard ones are described below.

- *Unit testing* is performed one program at a time, after changes have been made to each one, to ensure that each program executes as designed.
- *Function, component, or subsystem testing* brings a group of programs together for testing.
- *System testing* tests an entire system at once. Usually, several cycles of testing are required. A test that spans a few days can represent a full month of production cycles.
- *Stress testing* tests high volumes of transactions.
- *Regression testing* is the final test and involves a carefully selected set of test data. Any errors that occur in this stage are

corrected and the entire regression test is repeated, usually until a test is error free.

There are several preparatory steps to complete before testing can start. The test environment must be created. Test data, or test scripts, described earlier, must be written. Data must be converted. A method of recording problems and issues, which can take the form of a log, must be devised. As errors are corrected and retested, the log must be updated to reflect each of these activities. In well-run systems departments, all of these preparatory items are available for reuse on every project. Companies that are not as well prepared must schedule extra time to accomplish these steps.

An issue unique to y2k testing is the test environment. Many large companies can afford separate machines on which the system clock can be set forward to simulate the future dates required in y2k testing. Other companies without these facilities will have trouble replicating the exact environment without causing damage to production systems. Vendors have noticed this problem and are at the ready with machine time for y2k testing. Many of the larger accounting firms' consulting divisions offer machine cycles for testing purposes. See chapter 19, "Y2k Consultant," for more information on vendor services. Hardware and operating systems should be remediated first; then they should become part of the test environment. Application packages should be remediated and tested on the y2k-ready hardware and operating system platforms. This combination provides the most accurate test environment.

Who should perform the system test? Although the programmers who created the changes will probably be involved, they should not be the exclusive testers because of their built-in bias. It's best to have the owners of the system performing the system test instead of the programmers. Because the owners are the ultimate users of the system, they understand the business nuances that should be tested. They also must be able to use the new system, so the sooner they can start learning it, the easier the learning curve will be. Users will also notice any design issues that require modification before a system can be implemented.

Managers should make sure that testing isn't something that's done only if there is time after the regular daily work is done. It is important that employees be allowed to set aside or delegate their regular duties while they are on the test team so that they have ample time to conduct the tests in a thorough and accurate manner.

The degree of testing quality can vary. If the software can affect a person's health or is critical to our nation's defense, the level of testing must be at its highest: 100 percent defect-free. This means the test coverage must be thorough; every branch of program logic for every conceivable data item should be tested. If the system is a marketing system with quarterly analysis reports, the level of testing may not need to be as stringent. The testing quality should be a measure that is set and approved during the assessment phase of the project.

Change control is critical during testing. Testers should record every incident in the logs. Programmers should review the log results and modify programs as necessary. If a package is being tested, the vendor should be notified of problems in the software that are found during testing. Programs that were changed must be retested. The testing cycle is iterative, and good documentation must be kept as to what was tested, what was changed, and what was retested.

Regression testing is very important on y2k projects. Because so much code is changing, the chance of introducing additional errors is high. The Internal Revenue Service found this out the hard way when 1,000 innocent taxpayers received late payment notices. Also, a brokerage house accidentally deposited \$19 million in each of its client's accounts. These errors resulted from y2k changes that were not regression-tested. A regression test should be run until all new, unrelated errors are found and corrected.¹

Testing embedded systems can be tricky. Sometimes, the date can be forwarded in the embedded device, and test transactions can be entered. Often, the test environment cannot be replicated. In some cases, obtaining the vendor's compliance statement is the only cost-effective step to take. In other cases, that won't do at all. A combination of contacting the vendor and validating by testing is the best action when testing most embedded systems.

Recontamination is a big issue; companies have had to work hard at maintaining compliance once it has been achieved. Especially if spreadsheets have been remediated, it only takes one accountant to lapse into old habits to cause spreadsheets to revert to noncompliance. A combination of constant training, spot auditing, and careful bridging between compliant and non-compliant items will help to maintain y2k compliance in systems.

The testing portion of the project is very challenging and very hectic. Usually, the project is behind schedule, and the time allotted to testing is compressed to try to put the project back on schedule. Staff who have not been involved in the previous phases of the project are usually brought in for testing. These individuals can be from different departments, and a lot of coordination is required. As errors are discovered, they must be well documented. Expect the unexpected during testing; this is the whole reason for testing! One firm set the date forward on test machines, and when the users sat down to enter the test scenarios, all of their passwords had expired. The test had to be postponed, needless to say. But the event emphasized the importance of testing, contingency plans, and even tested contingency plans.² Although testing is hectic, fast-paced, and full of surprises, success will result from a project manager who stays organized and maintains his or her cool throughout this phase of the project.

CHAPTER 17:

IMPLEMENTATION

Several steps comprise the implementation phase of a y2k project. I'll briefly discuss the three components of implementation:

1. Training and documentation
2. Conversion and installation
3. Postinstallation troubleshooting



TRAINING AND DOCUMENTATION

Ideally, users are trained on a new system just before the system is installed. Users who have been active in testing the system will require the training earlier than that, prior to the testing phase.

A y2k project that modifies existing systems may not require much training. The user may have to enter a four-digit number in some cases where a two-digit number was entered previously. If that is the extent of the knowledge required after a system becomes y2k compliant, a class is probably not necessary. However, if a system has been replaced or upgraded, training will probably be required.

The same is true with system documentation for the users. Minor changes may be needed with systems that have been made compliant. Replacement systems and system upgrades will require new manuals. The documentation should be ready for distribution before installation or, before testing for users involved in the test phase.



CONVERSION AND INSTALLATION

The actual installation of the programs must be completed in a very short period of time. If the installation involves hardware, it is often installed in the wee hours of a Sunday morning when there is very little traffic on the systems and there is a chance to recover from any problems. The system components can be moved from the test environment to the production environment at that time as well.

There are literally hundreds of system components besides the programs to be installed. These include database definitions, screen layouts, file definitions, job control language, subroutines, and other items too numerous to mention. A good library management tool is essential in keeping these files straight throughout the project. The process of maintaining multiple versions of these system components is called *version control*. Keeping the files organized after all of the coding and testing changes is imperative to the quality control of the project.

Just before the first new program is executed in the production environment, the conversion routines must be executed. If the date expansion method was used to expand the date from a two-digit year to a four-digit year, then all history files and database contents must be converted to the new format. If the company's historical files are substantial, the conversion effort will probably span several weeks.

A y2k project can be huge for some companies. In some cases, it will be the largest project that a company has ever undertaken. For that reason, it's a good idea to split the project into subprojects that can be implemented at different times. It is better to implement the y2k project in phases instead of installing all changes in one big bang. In this way, the risk of system failure is reduced. However, if the systems interface with each other, temporary bridges will have to be built between systems that are y2k compliant and systems that are not y2k compliant. The project schedule can be devised with the interfaces in mind so that the number of required bridges is minimized.



POSTINSTALLATION TROUBLESHOOTING

Let's face it: No one is perfect. The first day after system installation can be hectic. There will probably be problems. It is a good idea to prepare for a larger number of calls in the end-user computing department on days when a system is installed. Training questions should be separated from system problems, and the problems should be communicated to the project team as soon as possible for rectification. After about a week, any unresolved problems should be turned over to the maintenance team who is now in charge of maintaining the system.

Companies that have faced massive y2k projects should put together a crisis team for the first few weeks (months?) of 2000 or until systems become stable. Everyone should be briefed on contingency plan actions and ready

to respond to a high number of problems. Details like a fully stocked refrigerator, extra paper, and reserved hotel rooms next to the office should be part of the crisis plan. Anyone who has offered y2k consulting or who sells software, hardware, or embedded systems should beef up staff as much as possible to handle problems in a crisis mode.

One last project task, which is hardly ever done because the project is late or the team is burned out, is to perform a project assessment. The project assessment provides a chance to review what was great and what was not so great about a project. One of the most important pieces of the assessment is to determine any differences between the estimated schedule and budget, and the actual project schedule and budget. Learning from these differences can make a project manager a much better estimator for the next systems project.

CHAPTER 18:

DATE MODIFICATION OPTIONS

Several options are available when changing code for y2k compliance:

- Date expansion method
- Windowing
- Date compression
- Encapsulation

The first two are the most common. In this chapter, I will assume that your existing date formats contain a two-digit year. In a recent luncheon speech to a Dallas y2k-users' group, City of Dallas' y2k-project manager John Raz said he has found programs dating from the 1960s that have only one position for the year.¹ The methods in this chapter should be adapted if that situation is encountered in your shop.



DATE EXPANSION

Expanding the date field from two digits to four digits is a simple concept. Most of the coding changes made using this method are made to the layout of the file and how it is defined to the program. The field length must be adjusted in the programs. A lot of the logic will not have to be changed, but the logic, and especially any calculations, should still be reviewed. Wherever the century value was used in the program, a change must be made to use the new four-digit year.

Although the coding changes will be minimal using this method, other changes are plentiful. Database definitions and descriptions must be changed. Job control language, used in a batch environment, must be reviewed. Screen layouts must be reviewed to see whether there is room for a four-digit year. There may be more than one date on any given screen, file, or database. For example: Shipping date, order date, and billing date could all be used on an order entry screen. Each date must be expanded.

This method affects the amount of storage media that is required by a company to process and store records. The expansion of many date fields can make it necessary to purchase additional tape or disk space to

accommodate the growth in databases or files. This expense must be factored into the budget for the project. Vendor lead times for ordering storage devices must be factored into the project schedule.

The date expansion method requires that existing data be converted to the new format. Conversion programs must be written, which will run one time, just before the changed programs are implemented, to convert the data from the old format to the new format. All historical data must be converted to the new format. The conversion of historical data alone can be a massive project for some companies.

The advantages and disadvantages of the date expansion method include:

- This method is the best long-term approach for systems that the company plans to run for a number of years.
- This method is best if you have the time and money to do it. This option is more expensive than others.
- The potential for errors is reduced as logic coding changes are minimal.
- This method is best for systems with dates that span many years.
- Systems with an enormous amount of historical data may not be the best candidates for this method because conversion would be required.
- Systems with limited screen space may not be able to fit the extra digits. These design questions must be answered before this method can be used.

Systems converted using the date expansion method will only work until the year 9999.



WINDOWING

Windowing is so named because only a range or window of dates will work in a program using this method. The date is still stored as two digits, with the century missing, so logic is put into the program to “guess” the century. There is usually a window of 100 years in which the program will return the correct answer.

Windowing code looks like this (in English):

```

IF THE DATE IS LESS THAN 50
  MAKE 20 BE THE CENTURY
OTHERWISE (IT WILL BE GREATER THAN OR EQUAL TO 50 SO)
  MAKE 19 BE THE CENTURY

```

This code will return correct answers for years between 1950 and 2049. The value 50 is called the pivot point, and is the standard pivot point used on most y2k projects for mainframe programs. Generally, the century value is stored in a temporary area during the program run and is not stored permanently in any file.

This method requires extensive testing. The concept is simple. Nevertheless, if programs work with multiple dates simultaneously and move them all over the place inside the program with poor documentation and poor naming standards, it is easy to accidentally introduce additional errors. These factors require that the programs be tested for correct results, especially if date computations are frequent.

The advantages and disadvantages of this method include:

- It is generally the cheapest method.
- This method is good for systems that you plan to replace soon but still need in the short term.
- It does not last forever and is not considered a permanent fix.
- It requires many coding changes that must be thoroughly tested.
- Bridging (temporary programs between systems that have been fixed and systems that have not) is minimized because file formats are not expanded.

A variation of windowing is the sliding window, in which the user can define and change the pivot point, or where the pivot point is calculated from the current year and slides forward into the future.



DATE COMPRESSION

Date compression, also called encoding, is somewhat like the opposite of date expansion. Instead of expanding fields from two digits to four digits, the date compression method crams four digits into two digits of storage. This method, however, is similar to date expansion in terms of the work that needs to be done. Some file formats and database descriptions will not

need to be lengthened using this method as they would with date expansion.

There are limitations to the compression of dates versus their storage and presentation, and this method must generally be used in conjunction with other methods when changing code.

The advantages and disadvantages are similar to the date expansion method. In addition, date compression lends itself well to automated tools.



ENCAPSULATION

Encapsulation, also called date translation, involves translating the date by aging it twenty-eight years. Why twenty-eight years? Because twenty-eight years ago, the calendar was identical to 1998. Thus, August 5 was a Tuesday in both 1970 and 1998. The translation is:

1997—1969
 1998—1970
 1999—1971
 2000—1972
 2001—1973

The encapsulation method works only with systems that are very self-contained and that do not have more than twenty-eight years' worth of data. The program logic is untouched. Before and after processing, date data is translated by aging it.

I have seen this suggestion mentioned with elevators that will have a y2k problem because of a chip that records maintenance times. If the date is set back twenty-eight years, the user of this method gains roughly thirty more years to make a decision about what they are going to do in the year 2000.

The encapsulation method, also called masking, is not very common or practical in most cases because of the integrated nature of today's systems. It is also considered a temporary fix. When it can be used, it can save millions, as in the case of Montgomery Mutual Insurance Company of Sandy Springs, Maryland. This insurer tricked its mainframe into thinking it was 1969. Before the users see the data, it is converted to the correct year using hypertext markup language conversion software. The little trick saved the company \$3 million, at least in the short run.²

CHAPTER 19:

Y2K CONSULTANTS

Several firms offer y2k-consulting services. I will list a random sample to give you an idea of what kinds of firms are offering y2k services and what services they are offering. I am not endorsing anyone.

Except for KPMG, the Big Five accounting firms' Web sites are silent about y2k services. A spokesperson for PricewaterhouseCoopers said that the y2k service has been "officially removed" from its product line. The firm still offers all other "normal" systems consulting services.

KPMG's Web site contains a paper about project management and describes its y2k services: assessment, renovation, testing methodology and support, program management office, and in-flight project reviews.

Harvey Goss, Partner at Goldstein, Golub, Kessler & Company, P.C., is offering y2k services at his New York firm. The services include an analysis of existing systems, searches, and assistance in locating replacement packages, and more. A special-issue, two-page y2k newsletter published by the partnership in 1997 generated a tremendous response from clients.

Another firm that offers y2k services is Gemko Consulting Group, a consulting group of the CPA firm Gaines, Metzler, Kriner and Co., LLP. After July 1, 1998, however, Gemko will stop taking y2k clients in order to maintain the quality of its existing commitments.

A few small firms are offering y2k consulting to their existing clients. A search for *CPA* and *year 2000* on the Internet turned up 1,500 to 3,000 hits compared to none a year ago. It is not too late to start offering y2k services, especially if you already offer technology services. A y2k analysis is a natural offshoot of technology planning and should at least be mentioned in these engagements. It is quite easy to sell a y2k engagement in conjunction with an assignment to do a three-year technology plan or an assignment to evaluate new software. As of summer 1998, I have completed several y2k plans for small companies.

Companies will pay just about anything for y2k services. Hourly rates for y2k work are running double or triple regular IT consulting rates. A consultant working for Geo. S. Olive & Co. in Indianapolis is billing at \$175 per hour for y2k work, per *Accounting Today*.¹ My research says that's low. However, some firms are not willing to gouge their current clients

with the higher rates. I try not to bill existing clients more than \$25 to \$50 per hour more than normal if I want that client for life, not just for y2k. If I think y2k will be the only project I will ever do for new clients, I quote them my highest rate, however.

A number of firms that are offering y2k consulting or are just interested in completing their internal projects have purchased my y2k practice aid entitled, *Guide to Y2k Consulting for Small Businesses*. It contains checklists, engagement letters, case studies, and an engagement program that should save CPAs time developing those items from scratch. It's sold with a legal disclaimer, like everything else designed to address y2k issues. If you are interested, go to <http://www.sandismith.com/y2kpract.html> for ordering information.

Many traditional systems development companies are offering y2k work. Cap Gemini's domestic y2k work makes up 12 percent of its revenue total. It has analyzed over two billion lines of code in the 100 projects it has performed to date. The limited-resource pool constrains Cap Gemini from doing more work. There isn't time to hire, train, and retrain people, plus they do not want to add too many people to their team as the y2k project shelf life remains short. Cap Gemini is interested in a complete outsourcing deal with clients, not just its y2k work.²

Cap Gemini also licenses its tools and methodology to clients as well as competitors. Its conversion tool finds dates in code, changes them, and tests them. It also generates bridge programs that are needed between two systems, one that is fixed and one that remains to be fixed, as portions of the project are completed and implemented.³

IBM has the largest staff of y2k consultants, which was expected to reach 4,200 by the end of 1997. That figure doesn't include 3,000 contractors plus another 3,000 individuals from partnering deals. It offers several y2k-related services:

- Complete outsourcing
- Conversion projects
- Seminars
- Tools and methodology
- Product support lines where customers can ask about compliance⁴

IBM is one of the few companies that has experience handling projects the size of y2k projects. It currently has 200 customers. To staff the massive amount of programmers needed, IBM has partnered with thirty businesses.⁵ Most consulting firms are finding that companies are late in responding to the y2k problem, but that their awareness is getting better. Most businesses created a budget and a plan in 1996. Ten percent of IBM's customers have progressed to implementation stages with portions of their code. Customers have been educated on the nature of consulting contracts regarding y2k work. Work is generally billed by time and materials and performed on a best-effort basis.⁶

Why are CPA firms, especially the large ones, keeping such a low profile on y2k consulting services? I can only conjecture, but a number of things come to mind:

- It's getting too close to December 31, 1999, to take any more work that could be completed in time, at least for large and medium-sized firms, where the money is better.
- The threat of litigation is scaring some firms, especially firms that are approaching y2k from an assurance standpoint. (I do not offer any assurance or any guarantees when I do y2k engagements, and clients accept those limitations.)
- Firms are overwhelmed by the work and unable to hire any more people.
- For small firms, it's early and they are still regrouping from a great tax season.

But one more thought may be the real answer. Think about how the y2k dilemma evolved. At first, we thought it meant fixing COBOL programs. Then we thought about embedded systems. Finally, we have discovered the y2k issue is really an enterprise-wide risk management exercise. Companies that signed clients with a lot of COBOL code changes may have used up all of their resources before they realized that they should have been offering a higher level of service. CPAs are especially good at weighing risks, but how many firms are approaching y2k consulting from a risk management perspective instead of an IT perspective?



NEED A CONSULTANT?

Should you use a consultant for all or a portion of your y2k project? Ask yourself the following questions before deciding:

1. Do you lack the expertise on your staff to competently complete a y2k project? (Be honest with yourself. I have talked to many CPAs who think they have their y2k projects handled. In fact, only about one-third of their projects are being handled and they do not know what is missing.)
2. Have you or your staff ever successfully completed a project the size of your y2k project? Was it on time and on budget?
3. Are you having trouble finding staff time to start a y2k project based on other business priorities?
4. Are you having trouble gaining executive sponsorship or budget money to start a y2k project?
5. Do you lack tools for analysis, coding, conversion, or testing y2k programs?

If the majority of your answers is yes, *probably* you should consider using a consultant.

If you have completed portions of the y2k project, determine your strengths and weaknesses in light of the tasks that remain. Consultants or tools can fill in gaps where in-house expertise is lacking.



CHOOSING A CONSULTANT

How should you choose a consultant? Another checklist will help to answer that question. Keep in mind this checklist is an ideal approach. Consultants who offer y2k services are busy, and most are signing clients so fast they have no time to work on proposals or pilots.

1. Decide which part of your project requires outside services.
2. Determine which solution providers have the skills that you lack. Make a list of solution providers.
3. Create a checklist to evaluate the vendors based on your requirements. The checklist will contain criteria that you have determined to be important to the project's success. For example,

- a.* The vendor has been in business for ten years or more.
 - b.* The vendor has completed similar projects successfully.
Ask for references.
 - c.* The project manager has more than fifteen years of experience in the industry and has been with the vendor for more than five years.
 - d.* Each member of the programming team has at least five years of similar experience on the same hardware and software platforms you are using.
 - e.* The vendor's solution is a fit for the company, both technologically and managerially.
 - f.* The vendor can get along with key management, customers, and other vendors.
 - g.* The programming team can follow company standards for modifying programs and creating documentation, or provides a methodology that will be followed.
 - h.* The vendor offers maturity, desire, and ability to do the project.⁷
- 4. Screen the list of vendors with a phone interview checklist or other method. Narrow the list down to five or fewer.
 - 5. Depending on the size of the project, you can choose to send a request for proposal (RFP) now or in lieu of the preceding step. The RFP should describe the hardware and software environment, the systems and programs, what needs to be done, and exactly what is needed from the vendor. Vendors will respond to the RFP, and the responses can be evaluated similarly to the phone interview answers. (This is a good idea for most technology projects, but not for y2k, where RFPs are frequently unanswered.)
 - 6. Narrow the vendor list to two or three.
 - 7. Look for a few small stand-alone systems to use as pilots. Also create an evaluation checklist. This will contain items such as the following:
 - a.* Quality of work
 - b.* Timeliness of work
 - c.* Organization, project management, and work schedules
 - d.* Conduct of vendor personnel: flexibility, friendliness, mesh with corporate culture, and responsiveness

- e.* Effectiveness of tools
- f.* Handling of issue resolution and change orders
- g.* Effective peer review mechanism
- h.* Formal change control processes
- i.* Client review procedures
- j.* Deliverables
- k.* Documentation
- l.* Cost⁸

8. Ask the vendor finalists to complete the pilot projects. Evaluate the vendors from that work and select a finalist.
9. Carefully negotiate a contract. Delegate items that you do not wish to be responsible for. Detail items, such as when programmers will work, what happens if the lead project manager is unable to complete the project, and the exact steps that are expected to be followed. Allow for escape clauses if problems set in. Write in due dates and expected deliverables so that progress can be accurately measured. If this is your first outsourcing contract, hire an attorney with specific outsourcing and information technology experience to write or review the contract. Many Fortune 500 companies have made costly legal mistakes on these agreements.

10. Continue to monitor and manage the project closely!

If your business is small, you will probably find that performing every step above is not cost-effective. The list can be narrowed down to suit a smaller or budget-minded business by using techniques such as keeping the list of vendors to a small size, skipping the RFP step, and creating checklists with a small number of key items.

CHAPTER 20:

TOOLS

Hundreds of vendors offer automated tools to speed the y2k-compliance process. The tools fall into a multitude of categories and perform various tasks. Tools are plentiful in some categories, such as analysis tools. In other cases, tools are nonexistent. Many of the tools are new and unproven as some vendors have gotten a late start. Others that address generic systems project management steps have existed for some time and have had the chance to build a reputation in the software industry.

Most companies have been extremely disappointed with y2k tools. A recent survey asked project managers to what extent tools have met their expectations. On a scale of 1 to 5, with 1 meaning “not met” and 5 meaning “exceeded,” the answer chosen most (35 percent) was 1: “not met.”¹

Companies that are considering tools should perform a thorough analysis to determine:

- Exactly what the tool does
- If the tool meets the business needs
- If the tool can do the job faster, cheaper, or better
- If the vendor will cooperate on a pilot project to test the tool’s effectiveness

It’s useful to break the myriad of tools available into a list by project phase. Not all of the tools used will be y2k specific; many will be useful on any large systems maintenance or development project.



ANALYSIS

One of the first steps in a y2k effort is to take an inventory of systems. If the business is small, a word processing table or spreadsheet will suffice as the tool of choice to capture inventory items and necessary characteristics. Medium-sized companies can quickly customize a desktop database to hold the information. Large companies may need a tool that automates the software inventory function. One y2k programmer built an integrated database to keep inventory, track vendor replies, and map interface

information in an Oracle database. I have built something similar in Microsoft Access (which is available as part of my y2k consulting guide; see my Web site for details). These tools do not have to be y2k specific. Having a well-documented inventory of systems and components is definitely an asset that can be used in multiple business functions.

Project management software should be used to schedule and track team members' participation on a project. This software is useful for estimating projects and later tracking the accuracy of the estimate against actual time logged. Some project management software incorporates the project methodology into the tool. I use Microsoft Project.

Date-identification tools sniff out the date fields contained in programs. Some are y2k specific; some are generalized date-finders. Some date identification tools simply look for dates. Others can do date calculations.

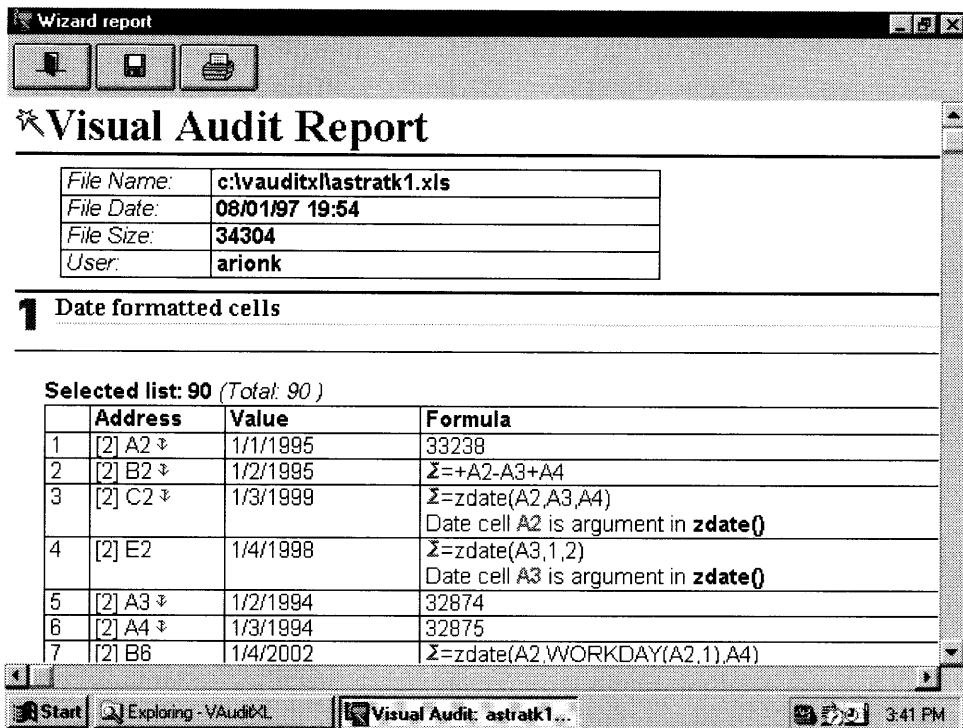
Other tools in the analysis category include data-impact analyzers. These tools trace the relationships between programs and systems and can generate statistics about the effect of a change on a program. Reports or graphs show where the problems are and provide estimates of how much it might cost to fix the programs. These tools can alert managers to the size of the problems in each system so that work can be prioritized and staffed.

Companies in the market for a tool should look for one that identifies dates and determines how they are used. The tool should trace the dates and the data contents of the date fields through the program. The tool's findings should be reported in a clear format (see figure 20.1).²

Analysis tools include:

1. *Platinum Technology, Inc.'s SystemVision Year 2000*. Analysts agree this is a solid tool. Developed by Adpac, it supports COBOL, PL/I, and Assembler languages. It finds date occurrences, analyzes their context, assigns a complexity rating, and writes the information into a database. Assumptions about costs and productivity are customizable in the reporting process. A what-if module aids decision making about various project approaches.³
2. *Isogon Corporation's TicToc*. This tool gives a number of date occurrences and the cost to fix them.
3. *Viasoft, Inc.'s VIA/SmartTest, VIA/Alliance, VIA/Recap, VIA/Insight, Estimate 2000 and VIA/SmartEdit*. For PCs and spreadsheets and other distributed systems issues, Viasoft has

FIGURE 20.1: A REPORT FROM VISUAL AUDIT SOFTWARE



a suite of tools called OnMark 2000 designed to take PC inventory, assess Excel spreadsheets, review client/server applications, and others.

4. *Mainware, Inc.'s HourGlass 2000*. Kemper Insurance's data administration manager, Mitch George, is happy with the results of this tool.⁴
5. *MicroFocus, Inc.'s Revolve/2000*. Kemper uses this tool for code analysis.⁵
6. *Global Software, Inc. Giles*.⁶
7. *AstraTek, Inc.* sells a product called *VisualAudit*. This product analyzes dates on Excel worksheets, Access, and Visual Basic programs. Stephanie Moore, an analyst at Giga Information Group, said the tool has been purchased by Wall Street types that maintain powerful spreadsheets.⁷ Check out AstraTek at <http://www.astratek.com>.



CONVERSION

Systems that will not be replaced or scrapped must be modified for y2k compliance. Conversion tools can inject the necessary date changes in the programming code. These tools are scanners, parsers, conversion systems, or multilanguage generators.⁸

In looking for a conversion tool, several factors must be considered. Speed and accuracy are two factors. How fast does the tool convert the code? How much of the necessary code is converted? Peritus Software offers a tool that can, by its claims, convert one million lines of code in one weekend. Eighty percent of the code that needs to be converted is converted.⁹ Formal Systems, Inc. says two statistics on accuracy are important to know:

1. False alarms (or false negatives), where code is changed that didn't need to be.
2. Misses (false positives), where code is not changed that should have been. They warrant their systems reengineering work to a 1 percent false-negative claim and a 5 percent false-positive claim.¹⁰

In a recent *Computerworld* article, Stephanie Moore, an analyst at Giga Information Group, advises companies to be wary of y2k vendors that promise more than about a 65 percent success rate with a conversion tool. She says she's found "hundreds of sleazy vendors" in the y2k marketplace.¹¹

You should find out whether the tool will perform the type of changes you want. For example, the y2k problem can be solved by field expansion, date compression, windowing, or other methods. How much of the process will actually be automated by the tool is another consideration. Some conversion tools plug into the analysis tools, streamlining the process.

Other types of tools that are useful during this phase of the project include version control software, data migration tools, bridging tools, and code preparation tools.

A few vendors that offer tools in this category include:

1. Compuware's Xpediter+, Xpediter/Xchange, File-Aid, Conversion Expert

2. Viasoft's VIA/Alliance, VIA/Valid Date, VIA/Insight, VIA/SmartEdit, VIA SmartTest, VIA/SmartTest-TCA, VIA/SmartDoc, VIA/Renaissance (Phillips Petroleum is using VIA/Insight on more than just its y2k project and finds Viasoft very available to handle questions and issues. Phillips is also using VIA/Alliance.)¹²
3. Peritus Software Services, Inc.'s AutoEnhancer/2000.
4. Computer Associates' CA-Realia II Workbench, CA-Visual Realia, CA-Verify, CALibrarian, CA-Panvalet, CA-Endecor, CA-PanAPT, CA-Datamacs/II, CA-Optimizer/II, CA-Migrate/COBOL
5. Platinum Technology, Inc.'s TransCentury Date Logic Generator, TransCentury Calendar Routines
6. IBS Conversions, Inc.'s IBS/Solution 2000 Tools¹³
7. Forecross Corporation converts C++ code
8. Software Emancipation Technology, Inc.'s Discover tool converts C and C++ applications¹⁴



TESTING

Testing the code is a huge part of the project, and there are a few tools to help with this phase. Many companies have created their own tools and test environments for systems development projects. Those that have will be ahead of the game.

Testing tools may handle various phases of testing, such as unit testing, module testing, system testing, stress testing, or regression testing. The buyer should determine the degree of automation, the platform, and the purpose of the tool. Tool types include date simulators, comparison utilities, date utilities, test data generators, and problem identification tools.

Vendors with tools in this category include:

1. Compuware's Xpediter+, XChange, QADirector, QA Hiperstation (Compuware has a good reputation and has been in the testing and debugging game for a while.)
2. Computer Associates' CA-Verify, CA-Datamacs/II, CA-InterTest, CA-InterTest/Batch
3. Princeton Softech, Inc.'s The Relational Tools¹⁵

4. Mainware, Inc.'s Hourglass 2000
5. Isogon Corporation's TICTOC
6. Softworks, Inc.'s HotDate 2000/Simulate
7. Prince Software, Inc.'s Simulate 2000
8. Viasoft, Inc.'s VIA/ValiDate
9. Platinum Technology, Inc.'s TransCentry Date Simulator
10. Advanced Software Products Group's Date/2000¹⁶



INTEGRATION TOOLS

At this last step of the project, the company must move the y2k-changed programs into *production*, which is the set of programs that the company runs on a daily basis. At this step, there are two versions of everything. Tools that manage version control will help in this phase. Bridging tools are also useful. A system that is ready to be placed into production may interface with others that haven't been changed yet. A bridging tool will create the necessary programs and files to bridge the differences between these programs.

Vendors with tools in this category include:

1. Computer Associates' CA-Endevor (This is a well-known version management tool.)
2. Princeton Softech, Inc.'s Version Manager (Dave Wilkins, field readiness manager at HOSTechnologies Geac, thinks this tool is great for reconciling y2k changes with ongoing maintenance.¹⁷)
3. Intersolv, Inc.'s PCVS Version Manager¹⁸

As the year 2000 gets closer, more tools will appear. Capers Jones estimates that 300 tools are already available, and that twenty to twenty-five new vendors are entering this market each month.¹⁹ Some of these tools are very immature and probably won't have the time to fully mature before the imminent deadline. Some advise to buy tools that will work for all project maintenance, not just the y2k project. Other tools are nonexistent for certain portions of the project or for certain platforms. Analysts advise to start with the vendors you already have relationships with. If you do have to develop new relationships, evaluate a company based on how good of a maintenance vendor it is, says Bruce Hall, research director at Gartner

Group.²⁰ Not all y2k tools will have a track record. When evaluating tools, consider the following:

- Does the vendor have a tool that supports the programming languages in your business?
- For a search tool, what percentage of y2k instances will the tool find?
- For a repair tool, what percentage of y2k instances will the tool fix?
- How many undetected instances will be left?
- Will the vendor fix new errors or incorrectly corrected code?
- What are the performance implications of the vendor's coding changes?
- How competent are the vendors' personnel?
- What guarantee is offered by the vendor?²¹
- Will the vendor do a test run? Evaluate a sample of the tools' results on your own code to see whether the tool will meet the goals you expect.
- What learning curve will it take for your staff to use the tool effectively?

How much are companies spending on tools? Mid-sized companies have budgeted \$500,000 to \$750,000, which includes tools and training, or 5 to 10 percent of their y2k-project costs.²² (Since my source for this figure was partially funded by advertising dollars, be wary.) All tools should be cost-justified before a decision is made. If the tool is expected to save money and time, it should be purchased. If not, do not purchase it.

CHAPTER 21:

FINAL TIPS

In this chapter, I'll present sources for further reading, a call to spread the word, and final tips on protecting yourself from y2k failures that are out of your control.



SOURCES FOR FURTHER READING

The search for best practices on y2k projects will continue for years to come. Two online discussion groups frequented by individuals who have the daring job of assuring that the rest of us experience the century change unscathed are worth mentioning as a source of tips from the front line.

They are:

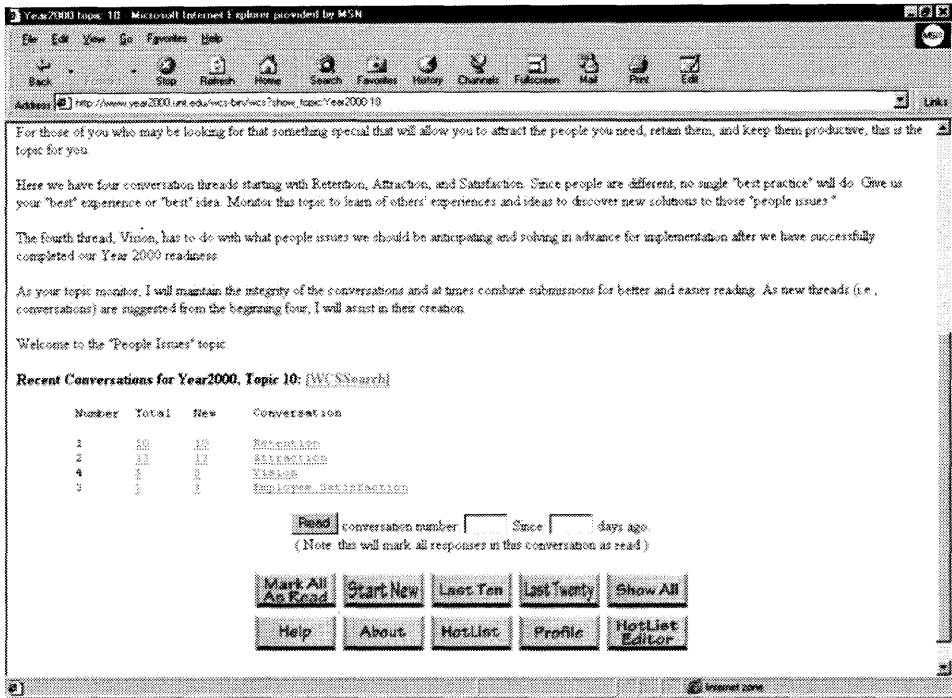
- Best Practices Web site (see figure 21.1) sponsored by the Society for Information Management (SIM) Year 2000 Working Group (Find it at <http://www.simnet.org>.)
- The Usenet news group: comp.software.year-2000 (see figure 21.2)

Another leading source of information on the y2k specifically for CPAs is the AICPA Web site. You can use the search engine, enter “year 2000,” and find multiple postings of interest. A y2k page lists relevant links to the major sites devoted to dispensing y2k information. It's at <http://www.aicpa.org> (see figure 21.3).

One more Web site that provides y2k information for CPAs is my own, at <http://www.sandismith.com>. In the y2k section, you can download PowerPoint presentations, review checklists, and read articles. You can also see a sampling of the training material that is available on the y2k subject. The AICPA offers a video-based CPE course, which includes this book, that provides eight hours of continuing professional education (CPE).

If you ask the question, “Who is the leading authority on year 2000?” most individuals in the know would say Peter de Jager. If you had time to visit only one Web site, the Web site offered by Peter de Jager's company and The Tenagra Corporation should be the one. You can find it at <http://www.year2000.com> (see figure 21.4).

FIGURE 21.1: A THREAD ON SIM'S YEAR 2000 WORKING GROUP'S BEST PRACTICES WEB SITE



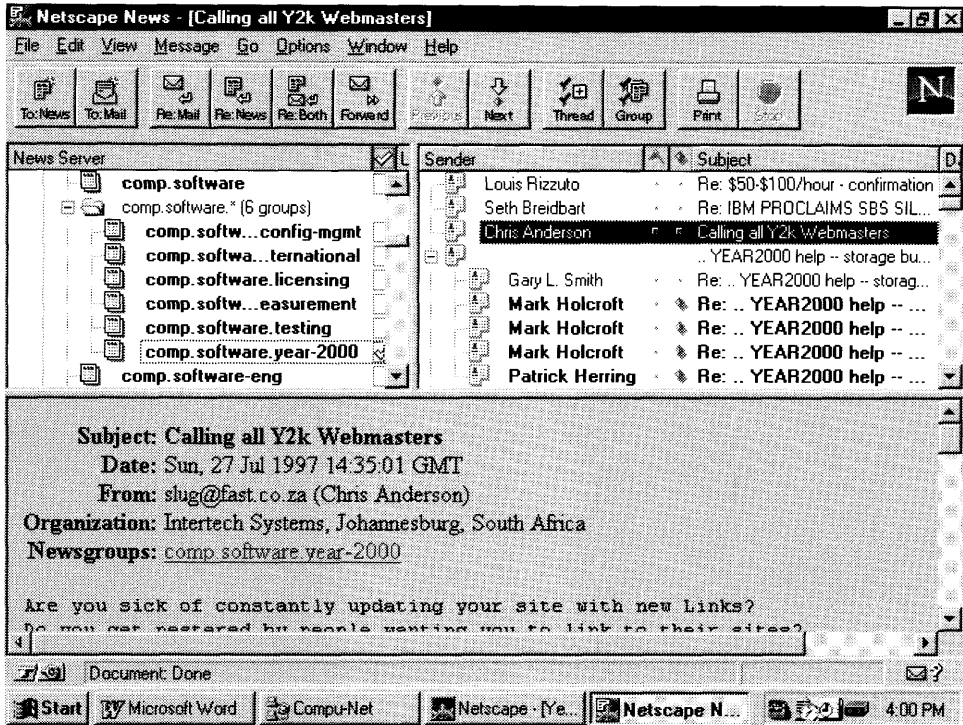
I know CPAs love checklists, and I have found some good ones at the Small Business Administration's Web site, <http://www.sba.gov/y2k/morechecklists.html>. Follow the checklist link to find great offerings from Stanford University and E-Commerce.com.

I have found some great stuff at Herbert Elliott's y2k site, the Mother of All Year 2000 Sites. It's at <http://pw2.netcom.com/~helloitt/00.htm>. I think it was from this site I found a fairly good paper on y2k for small-business owners at <http://millennia-bcs.com/summary.htm>.

If you have time to read only one more book besides this one, try William Ulrich's and Ian Hayes' *The Year 2000 Software Crisis: The Continuing Challenge*. This book focuses on business solutions, instead of personal extremism which unfortunately is the focus of others in print.

If you have time to read only one y2k magazine, the *Year 2000 Journal* should be it. Online you can find it at <http://www.y2kjournal.com>. Although some articles are still about awareness, there are at least one or two real jewels in each issue.

FIGURE 21.2: A THREAD ON USENET'S COMP.SOFTWARE.YEAR-2000 NEWS GROUP



There are plenty of y2k conferences available. I have not personally attended any because of my limited budget (hey, I'm a writer, for crying out loud) but I have seen offerings from Software Productivity Group (<http://www.spgnet.com>) and DCI (<http://www.dci.com>).

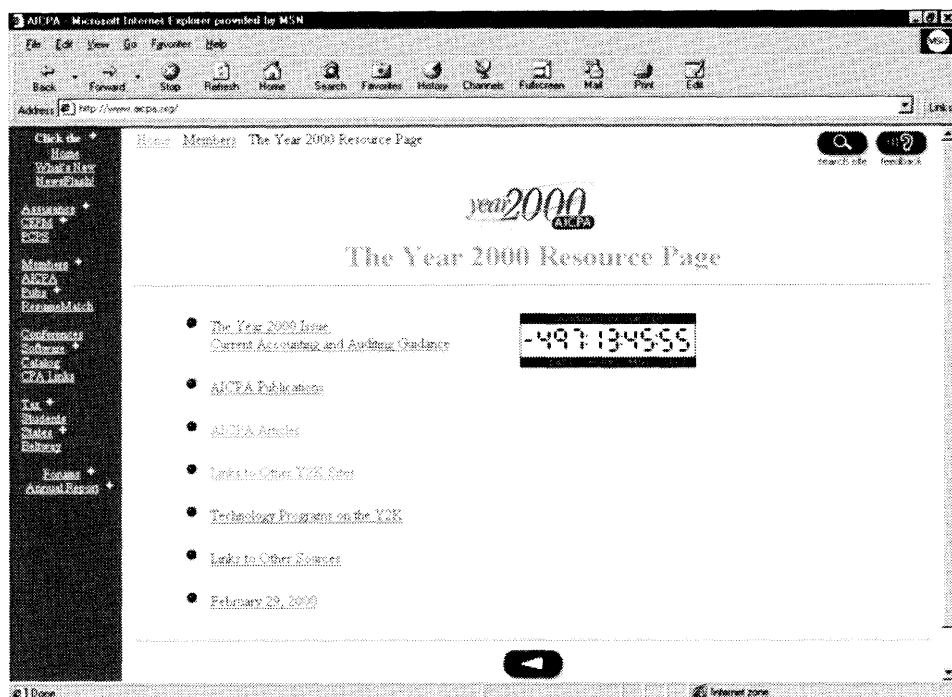


SPREAD THE WORD

Individuals who can accurately explain the problem to others will help to raise the awareness level of the y2k problem. If everyone is working to prevent the problem, then the impact will not be as devastating as some predict.

As CPAs, one of our core competencies is the trust that others have in us. We can enhance that trust by participating in the y2k efforts of our employers and our clients. Ensure that your employers and clients understand the issues and are acting to solve the problem. You may well be saving a few companies from financial disaster by spreading the word.

FIGURE 21.3: THE AICPA'S SITE PROVIDES EXCELLENT COVERAGE OF THE Y2K PROBLEM



PROTECTING YOURSELF FROM THE FAILURES OF OTHERS

Are there some companies that you depend on that are not addressing their y2k problem? If you have no control over the situation, the best way to be prepared is to create a contingency plan.

If you think your employer will not be ready, and you have done everything in your position to stir the leaders into action, you will want to decide what you will do as the date approaches. (Print your resume *before* the LAN goes down on 1/1/00.) One great way to find out how a y2k project is *really* going is to ask the staff programmers. They will be the first ones to know whether the project is failing or not. The project manager will also have a good idea and can make or break a project with his or her actions.

If you think a vendor that you do business with will not be ready, research competitors before the date approaches and decide on a contingency plan. If there are no competitors, try to time purchases and actions so that the y2k problem will have minimum personal impact. There will not have to

FIGURE 21.4: THE Y2K INFORMATION CENTER



be a snowstorm in December 1999, for there to be a run on products and services in certain businesses.

If you have investments in businesses, you will want to stay current on the status of their y2k efforts in order to protect your investments.

Hopefully, you will not have to use your contingency plans, but it is always better to be overprepared than underprepared.

CHAPTER 22:

LIFE AFTER THE YEAR 2000

The y2k problem will be with us well past January 1, 2000. Once we make it through that interesting first week, the next milestone, February 29, 2000, is around the corner to offer another chance to validate our readiness. Lawsuits will just be gathering momentum and are expected to continue into 2005.

The y2k problem is a chance for nonprogrammers to better understand how long systems changes will take. If a basic assumption is not followed, such as taking into account the year 2000, the costs can be enormous. What if we decided to do a similar assumption-shattering project? For example, what if we decided to expand the area code to four digits? An area code expansion project would not take as long as a y2k project (there would be no need to examine calculations, for example), but it would be a big project, nonetheless. Every phone number field would have to be found and expanded. If a basic assumption is changed, the project can be massive.

Is there anything we can do to prevent these problems or to circumvent the length of correction? It is not cost-effective to program every bell and whistle that could arise in the next twenty years. Hopefully, newer technologies will be designed so that maintenance is minimized for those types of changes. I do not think that they exist yet, although some of the object-oriented technologies move in that direction.

Many people will ponder the y2k problem for years. It has and will be the subject of many dissertations, papers, articles, books, CD-ROMs, digital video disks (DVDs), and whatever format will be popular in the future.

For now, however, you have less than 500 days until 1/1/00. . . .

APPENDIX A:

SYSTEMS INVENTORY CHECKLIST



HARDWARE/OPERATING SYSTEMS

- ☐ Mainframe hardware, operating systems, utilities
- ☐ LAN/WAN hardware, operating systems, utilities
- ☐ Minicomputer hardware, operating systems, utilities
- ☐ AS/400 hardware, operating systems, utilities
- ☐ Client/server hardware, operating systems, utilities
- ☐ Other platforms hardware, operating systems, utilities
- ☐ PC hardware, operating systems, utilities
- ☐ Hardware at remote sites, including employees' homes and backup sites
- ☐ Laptop/notebook hardware, operating systems, utilities
- ☐ Handheld devices



APPLICATION SYSTEMS

All business systems:

- | | | |
|--|---|---|
| <input type="checkbox"/> manufacturing | <input type="checkbox"/> banking | <input type="checkbox"/> financial reporting |
| <input type="checkbox"/> sales | <input type="checkbox"/> marketing | <input type="checkbox"/> project management |
| <input type="checkbox"/> order entry | <input type="checkbox"/> legal | <input type="checkbox"/> time and billing |
| <input type="checkbox"/> distribution | <input type="checkbox"/> contracts | <input type="checkbox"/> research and development |
| <input type="checkbox"/> accounting | <input type="checkbox"/> fixed assets | <input type="checkbox"/> tax |
| <input type="checkbox"/> payroll | <input type="checkbox"/> capital appropriations | <input type="checkbox"/> process control |
| <input type="checkbox"/> human resources | <input type="checkbox"/> general ledger | <input type="checkbox"/> security |
| <input type="checkbox"/> purchasing | <input type="checkbox"/> data warehouses | <input type="checkbox"/> point of sale |
| <input type="checkbox"/> shipping | <input type="checkbox"/> databases | <input type="checkbox"/> retail |
| <input type="checkbox"/> inventory | <input type="checkbox"/> decision support | <input type="checkbox"/> backup and disaster recovery systems |
| <input type="checkbox"/> accounts receivable | <input type="checkbox"/> forecasting | |
| <input type="checkbox"/> accounts payable | <input type="checkbox"/> planning | |



EXTERNAL INTERFACE SYSTEMS

- ☐ Electronic Data Interchange (EDI) systems (list of trading partners)
- ☐ External ordering systems (list of all vendors transmitting data)
- ☐ Extranets
- ☐ Any systems that originate in the company and connect electronically with other businesses

- ☐ Vendors and suppliers that provide your company with products and services, such as utilities, office supplies, equipment, or your landlord



USER AND DESKTOP APPLICATIONS

- ☐ Spreadsheets
- ☐ Macros
- ☐ Databases
- ☐ Contact managers
- ☐ Schedulers
- ☐ Email packages
- ☐ Messaging systems
- ☐ Groupware
- ☐ Backup systems
- ☐ Customized report writers
- ☐ Any fourth-generation programs not maintained by systems department
- ☐ User-developed applications
- ☐ Personal digital assistants
- ☐ Calculators
- ☐ Pagers
- ☐ Wristwatches



OFFICE SYSTEMS

- ☐ Credit card systems
- ☐ Copiers
- ☐ Fax machines
- ☐ Telephone systems
- ☐ Answering machines
- ☐ Voice mail systems
- ☐ Mobile telephones
- ☐ TVs
- ☐ VCRs and other video projection equipment
- ☐ Time clock systems
- ☐ Postage machines
- ☐ Coffeemakers
- ☐ Microwave ovens
- ☐ Backup systems
- ☐ Scheduling systems
- ☐ Scales



BUILDING SYSTEMS

- ☐ Heating and cooling systems
- ☐ Thermostat controls
- ☐ Door locks
- ☐ Lighting systems
- ☐ Cable TV systems
- ☐ Fire control systems
- ☐ Security monitoring systems
- ☐ Environmental safety systems
- ☐ Landscape and sprinkler systems
- ☐ Safes and vaults
- ☐ Card entry systems
- ☐ Elevators, escalators, and lifts
- ☐ Parking systems (gate entry, ticketing and meters)
- ☐ Other building maintenance systems
- ☐ Vending machines

**SPECIALIZED INDUSTRIES**

- ☐ Ticketing systems
- ☐ Health equipment for hospitals and physicians' offices (defibrillators, pacemakers, others)
- ☐ Automated teller machines for banks
- ☐ Process control systems for manufacturing plants
- ☐ Traffic systems for transportation industries (traffic lights, air traffic control, global positioning systems (GPS), railroad switching, radar systems, parking meters)
- ☐ Services and utilities (water and sewage, electric, flood and disaster control, waste disposal, natural gas)
- ☐ Automated factories
- ☐ Communications industry (entertainment satellites)

APPENDIX B:

SAMPLE PROJECT PLAN



PROJECT INITIATION

1. Identify the executive sponsor and/or steering committee. Set meeting times, dates, and deliverables.
2. Identify the project manager and project organization.
3. Change the acquisition/purchasing policy to assess future purchase of y2k-compliant goods and services only. Assess and change other policies, such as credit and collections, mergers and acquisitions, insurance coverage, and other y2k-affected policies.
4. Determine the project methodology and project management tools.
5. Create an initial project plan and begin the approval process.
6. Create project standards.
7. Determine whether a project office is appropriate and, if needed, create one.
8. Hire staff.
9. Begin an internal awareness campaign. This may include sending a newsletter or conducting training sessions with all employees in the organization.
10. Complete a project schedule and budget for the assessment stage.



ASSESSMENT

1. Determine whether the organization will benefit from automated tools in this step. If so, analyze and purchase them.
2. Select and acquire inventory tracking software.
3. Determine whether products and services being sold are y2k compliant and take any necessary actions, including contacting a lawyer. Suppliers may need to be sent correspondence to determine the compliance of parts and materials used.
4. Begin an inventory of systems.
5. Send vendor notification letters for packaged systems.

6. Send letters to customers and suppliers with systems that interface with your systems.
7. Complete the inventory.
8. Review contracts with vendors to determine options for packaged and purchased systems.
9. Begin an analysis of options with each major system listed in the inventory.
10. Determine whether each system will be fixed, replaced, updated, or scrapped.
11. For systems that will be fixed, determine the date conversion method and complete the analysis and design work for these systems, including the creation of test plans.
12. For systems that will be replaced or updated, create new projects outside the scope of this project, and add reporting requirements between project managers to keep up with schedule requirements.
13. For systems that are supposedly y2k ready, determine whether testing would be prudent and should be added to the project plan.
14. Send letters to employees notifying them of systems (like spreadsheets) that are out of the scope of the project and what they need to do.
15. Prioritize the systems and schedule the various initiatives.
16. Complete the project schedule and budget for the renovation stage.
17. Create a contingency plan for each system listed in the inventory.



RENOVATION

1. Purchase additional hardware if needed for testing.
2. Acquire hardware and operating system patches for remediation.
3. Acquire software packages for remediation.
4. For systems that will be eliminated, pull the plug.
5. For systems that will be fixed, select and acquire the tools that will be used during conversion.
6. Hire staff to perform the program changes.

7. Train the staff on project standards.
8. Begin the program conversion.
9. Conduct unit testing.
10. Test and remediate hardware platforms and operating systems.
11. Set up the test environment.
12. Install packages into test environment.
13. Create test data and revalidate the test plan.
14. Conduct system testing. Maintain and resolve a problem log and an issues log. Maintain version control.
15. Communicate testing results to package vendors.
16. Work with electronic business partners on interfaces.
17. Conduct a regression test.
18. Begin user and systems documentation.
19. Monitor related projects, in which systems are being replaced or upgraded.
20. Report project status to all of the auditors, regulatory agencies, and managers that are tracking this project.
21. Complete a project schedule and a budget for the implementation stage.
22. Create a crisis plan.



IMPLEMENTATION

1. Select and acquire tools for the implementation phase.
2. Complete system and user documentation.
3. Conduct user training.
4. Ramp up end-user computing staff.
5. Convert historical data and run all conversion programs.
6. Install bridge routines and install y2k programs and packages into production. Maintain version control.
7. Troubleshoot new implementation.
8. Complete a project assessment report.

ENDNOTES

SECTION I: THE YEAR 2000 DILEMMA



CHAPTER 1: IT'S BIGGER THAN A BREADBOX

- ¹Holman W. Jenkins Jr., "Turns Out the Year 2000 Problem is Just the Beginning," *The Wall Street Journal*, May 6, 1997, page A23.
- ²Rick Telberg, "Start Worrying: The Year 2000 Bug Is For Real," *Accounting Today*, June 2–15, 1997, page 6.
- ³"The Millennium Challenge," From the Merrill Lynch Forum, <http://www.ml.com/forum/millen.htm>.
- ⁴Rich Miller, "Greenspan Urges Focus on 2000 Computer Glitch," *USA Today*, February 26, 1998, page 1.
- ⁵See an interesting paper explaining his ideas at <http://www.yardeni.com>.
- ⁶James L. Cassell, "Gartner Group Testifies on Year 2000 Problem before House Committee on Science," <http://gartner3.gartnerweb.com/gg/static/itjournal/testimony1.html>, November 4, 1997.
- ⁷Michael J. Mandel and others, "Zap! How the Year 2000 Bug Will Hurt the Economy," *Business Week*, March 2, 1998, page 93.
- ⁸Amy Malloy, "Beat the Clock!," *Computerworld*, January 6, 1997, page 67.
- ⁹Lee Gomes, "Companies Estimate 'Year 2000' Computer Costs," *The Wall Street Journal*, April 9, 1998, page B8.
- ¹⁰Paul Strassmann, "Numbers Add Up to a Bigger Year 2000 Disaster," *Computerworld*, June 9, 1997, page 78.
- ¹¹Capers Jones, "The Global Economic Impact of the Year 2000 Software Problem," Software Productivity Research, Inc., January 23, 1997, page 45, 58.
- ¹²Steven Levy and Katie Hafner, "The Day the World Shuts Down," *Newsweek*, June 2, 1997, page 53.
- ¹³Rick Telberg, "Start Worrying: The Year 2000 Bug Is For Real," *Accounting Today*, June 2–15, 1997, page 6.
- ¹⁴Steven Levy and Katie Hafner, "The Day the World Shuts Down," *Newsweek*, June 2, 1997, page 53.
- ¹⁵"Briefs: Probabilities of Year 2000 Damage," *Computerworld*, April 20, 1998, page 43.



CHAPTER 2: DON'T TAKE IT PERSONALLY, BUT. . .

- ¹Thomas Hoffman and Robert L. Scheier, "Year 2000 Bug Bites Credit Cards," *Computerworld*, April 14, 1997, page 2.

- ²Holman W. Jenkins, Jr., "Turns Out the Year 2000 Problem Is Just the Beginning," *The Wall Street Journal*, May 6, 1997, page A23.
- ³Patrick L. Porter and Deborah Radcliff, "Time Bomb!," *Software Magazine*, March 1997, page 32.
- ⁴Ibid.
- ⁵Wendy S. Mead, "Chase Projects \$250M Cost to Avert Year-2000 Glitches," *American Banker*, April 11, 1997.
- ⁶"Special Edition: The Year 2000," *Chase Week*, April 6, 1997.
- ⁷Matt Hamblen, "Y2k Shortcomings May Shutter Some Banks," *Computerworld*, February 16, 1998, page 1.
- ⁸Thomas Hoffman, "Securities Group to Chief Execs: Get Involved," *Computerworld*, May 26, 1997, page 1.
- ⁹Bob Violino, "Ready for 2000?—Securities Firms form Group to Plan for Year 2000 Compliance," *Information Week*, December 9, 1996, at <http://www.techweb.com>.
- ¹⁰Laura DiDio, "Wall Street to Test Y2k Readiness," *Computerworld*, February 9, 1998, page 1.
- ¹¹Steve Levy and Katie Hafner, "The Day the World Shuts Down," *Newsweek*, June 2, 1997, page 53.
- ¹²"USA Today's Market Scoreboard: Street Talk," *USA Today*, May 21, 1998, page 5B.
- ¹³Matt Hamblen, "Utilities' Systems Aren't Likely to Be Fixed by 2000," *Computerworld*, May 18, 1998, page 3.
- ¹⁴Kim Girard and Robert L. Scheier, "Telcos Lag on Year 2000, Analysts Warn," *Computerworld*, November 11, 1996, page 8.
- ¹⁵Steve Levy and Katie Hafner, "The Day the World Shuts Down," *Newsweek*, June 2, 1997, page 53.
- ¹⁶Michael Mecham, "Industry Tackles Year 2000 Problem Systematically," *Aviation Week & Space Technology*, May 26, 1997, page 61.
- ¹⁷Stewart Deck, "Year 2000 Syndrome Strikes Again," *Computerworld*, July 22, 1996, page 8.
- ¹⁸Joe Gwinn, "The GPS Week 1024 Rollover," <http://www.navcen.uscg.mil/gps/ggeninfo/gps1999.txt> or <http://www.cominfo.co.uk/y2k/gps.htm>.
- ¹⁹"Street Talk: Investors Must Keep 2000 Bug in Mind," *USA Today*, May 21, 1998, page 5B.
- ²⁰Thomas Hoffman, "Year 2000: Hospitals Diagnose Themselves in Critical Condition," *Computerworld*, March 2, 1998, page 24.
- ²¹Mitch Betts, "A Costly Specter," *Computerworld*, May 19, 1997, page 28.
- ²²Ann K. Coffou, "Year 2000 Risks: What Are the Consequences of Technology Failure?" Statement of Hearing Testimony, http://www.house.gov/science/coffou_3-20.html.

- ²³Mitch Betts, "A Costly Specter," *Computerworld*, May 19, 1997, page 28.
- ²⁴Gary H. Anthes, "Feds Face Year 2000 Crisis," *Computerworld*, April 22, 1996, page 1.
- ²⁵Committee on Government Reform and Oversight, "Year 2000 Computer Software Conversion: Summary of Oversight Findings and Recommendations," House Report 104-857, September 27, 1996, <http://www.house.gov:80/reform/year2000.htm>.
- ²⁶Joel C. Willemssen, "Year 2000 Computing Crisis—Strong Leadership Today Needed to Prevent Future Disruption of Government Services," Statement Before a House Subcommittee, February 24, 1997, <http://www.access.gpo.gov>.
- ²⁷Gary H. Anthes, "Double-Check Your Bridge to the Year 2000," *Computerworld*, November 17, 1997, page 37.
- ²⁸M.J. Zuckerman, "Government Earns an 'F' for Fixing Year 2000 Woes," *USA Today*, June 3, 1998, page 6A.
- ²⁹M.J. Zuckerman, "Clinton Orders Attack on Year 2000 Computer Bug," *USA Today*, February 5, 1998, page 2A.
- ³⁰Matt Hamblen, "Senate Panel Pushes for Repairs," *Computerworld*, May 11, 1998, page 41.



CHAPTER 3: STRICTLY BUSINESS

- ¹Chuck Mumm, Dallas Water Utilities, City of Dallas, from an interview with me in the summer of 1997. He was primarily referring to XT PCs.
- ²Laura DiDio, "Y2k Snag Hides in PC Hardware," *Computerworld*, April 13, 1998, page 24.
- ³A white paper from Compaq Computer Corporation, found at http://www.wa.gov/dis/2000/survey/dt_hard/compaq.htm, dated January 1996.
- ⁴A white paper from Compaq Computer Corporation, found at http://www.wa.gov/dis/2000/survey/dt_hard/compaq.htm, dated January 1996.
- ⁵Terry Slade, "The Year 2000 Problem on PC Systems," Micro Firmware Technical Support Web pages, found at <http://www.firmware.com/pb4ts/year2000.htm>, February 26, 1997.
- ⁶A letter from Apple Computer, Inc. found at http://www.wa.gov/dis/2000/survey/dt_hard/apple.htm, dated July 8, 1996.
- ⁷A white paper from Compaq Computer Corporation, found at http://www.wa.gov/dis/2000/survey/dt_hard/compaq.htm, dated January 1996.
- ⁸"The Year 2000 Problem: How It Relates to Personal Computers and Personal Computer Software," found at <http://www.microsoft.com/cio/articles/year2000final.htm>, September 19, 1996.
- ⁹The Microsoft Year 2000 Resource Center: Excel 97, found at <http://www.microsoft.com/year2000/>.

- ¹⁰“What Does the Year 2000 Mean to Corel Users?” found at <http://www.corel.com/2000.htm>.
- ¹¹“Lotus and the Year 2000,” found at <http://www.lotus.com/uk/220e.htm>, March 26, 1997.
- ¹²“The Year 2000 Problem: How It Relates to Personal Computers and Personal Computer Software,” found at <http://www.microsoft.com/cio/articles/year2000final.htm>, September 19, 1996.
- ¹³“What Does the Year 2000 Mean to Corel Users?” found at <http://www.corel.com/2000.htm>.
- ¹⁴“Lotus and the Year 2000,” found at <http://www.lotus.com/uk/220e.htm>, March 26, 1997.
- ¹⁵From a speech, “Year 2000 Issues with Distributed Computing,” by Ed Hatchett, Year 2000 Manager of Alltel, presented to the DFW DAMA Y2k Group on March 20, 1998.
- ¹⁶Capers Jones, “The Global Economic Impact of the Year 2000 Software Problem,” Software Productivity Research, January 23, 1997, page 22.
- ¹⁷Capers Jones, “The Global Economic Impact of the Year 2000 Software Problem,” Software Productivity Research, January 23, 1997, page 20.
- ¹⁸Warren S. Reid and Steven Brower, “Beyond Awareness: Ten Management and Ten Legal Pitfalls Regarding the Year 2000 Computer Problem That You May Not Have Considered Yet!,” <http://www.year2000.archive.com/beyond.html>, page 16.
- ¹⁹William Farrell and Charles E. Phillips, “Year 2000 Watch: Survey at Year 2000 Conference,” Morgan Stanley U.S. Investment Research, April 25, 1997.
- ²⁰Ann K. Coffou, “Year 2000 Risks: What Are the Consequences of Technology Failure?” Statement of Hearing Testimony, http://www.house.gov/science/couffou_3-20.html.
- ²¹Steve Rosenbush, “Year 2000 Bug Threatens Phone Service,” *USA Today*, May 19, 1998, page B1.
- ²²Kristi Essick, “IBM Rolls Out Several Net Support Products,” *Computerworld*, June 2, 1997, page 56.
- ²³Thomas Hoffman and Mitch Betts, “Year 2000 Glitches May Be Everywhere,” *Computerworld*, May 19, 1997, page 1.



CHAPTER 4: INSIDE AND OUTSIDE THE ORGANIZATION

- ¹“What’s News—Business and Finance,” *The Wall Street Journal*, July 25, 1997, page 1.
- ²Robert L. Scheier and Thomas Hoffman, “Big Customers Press Suppliers on Year 2000,” *Computerworld*, March 3, 1997, page 17.
- ³Michael Mecham, “Industry Tackles Year 2000 Problem Systematically,” *Aviation Week and Space Technology*, May 26, 1997, page 61.

⁴Robert L. Scheier and Thomas Hoffman, "Big Customers Press Suppliers on Year 2000," *Computerworld*, March 3, 1997, page 17.

⁵Thomas Hoffman and Robert L. Scheier, "Supply Chain Leaders Push for Compliance," *Computerworld*, March 3, 1997, page 1.

⁶Thomas Hoffman and Robert L. Scheier, "Supply Chain Leaders Push for Compliance," *Computerworld*, March 3, 1997, page 1.

⁷Matt Hamblen, "Telcos Have Banks on Hold," *Computerworld*, April 13, 1998, page 6.



CHAPTER 5: 2000 IS A LEAP YEAR

¹Jerome T. Murray and Marilyn J. Murray, "The Year 2000 Computing Crisis," McGraw-Hill, 1996, page 5.

²Ibid.

³Ibid.

⁴Mark Olsen, "A New Twist on Year 2000," letters, *Computerworld*, April 7, 1997, page 34.

⁵John Figura, "Oops! There Is a 2/29/2000," letters, *Computerworld*, April 14, 1997, page 34.



CHAPTER 6: AND NOW FOR THE SILVER BULLET

¹"Program Offers Quick Fix to Year 2000 Glitch," *The Dallas Morning News*, June 24, 1997, page 16D.

²Thomas Petzinger Jr., "Bob Bemer Aims 'Silver-Plated Bullet' at Year 2000 Problem," *The Wall Street Journal*, June 20, 1997, page B1.

SECTION II: YEAR 2000 ISSUES



CHAPTER 7: AWARENESS AND THE ANTICS THAT FOLLOW

¹Charles E. Phillips and William E. Farrell, "Technology: Enterprise Software; Y2k Watch: Digital Plagues Oozes Across Planet," *Morgan Stanley U.S. Investment Research*, March 26, 1997, page 8.

²"Summary of Results, Year 2000 Survey," found at <http://www.year2000.com/archive/y2ksurvey.html>, updated May 9, 1997.

³Ibid.

⁴Robert L. Scheier, "Many Year 2000 Projects Still at Starting Gate," *Computerworld*, April 14, 1997, page 24.

⁵Tom Diedrich, "Top U.S. Companies Finally Take Y2k Problem Seriously," *Computerworld*, April 13, 1998, page 16.

⁶Rodney Ho, "Many Small Business Owners Shrug at Year 2000 Problem," *The Wall Street Journal*, June 2, 1998, page B2.

⁷Robert L. Scheier, "The Road to 1/1/00," *Computerworld*, December 9, 1996, page 83.



CHAPTER 8: ACCOUNTING ISSUES

¹AICPA, *The Year 2000 Issue: Current Accounting and Auditing Guidance*, Product No. 022503; price to members \$11, non-members \$13.75. To order, call (800) 862-4272. This publication is also available from the AICPA website: www.aicpa.org/members/y2000/intro.htm.

²Stephen Barlas, "SEC Revises Year 2000 Guidance," *Management Accounting*, March 1998, page 12.

³Michael Diamond, "SEC to Prod Firms on Year 2000 Costs," *USA Today*, June 11, 1998, page 2B.

⁴Howard Gleckman, "Hey, I Owe 99 Years in Back Taxes," *Business Week*, February 23, 1998, page 119.

⁵AICPA, *The Year 2000 Issue: Current Accounting and Auditing Guidance*, Product No. 022503; price to members \$11, non-members \$13.75. To order, call (800) 862-4272. This publication is also available from the AICPA website: www.aicpa.org/members/y2000/intro.htm.

⁶Jeff Jinnett, "Legal Issues Concerning the Year 2000 'Millennium Bug,'" <http://www.comlinks.com/legal/jjin1.htm>.



CHAPTER 9: LEGAL ISSUES

¹Susan Adams, "The Bug Bar," *Forbes*, July 28, 1997, page 45.

²Capers Jones, "The Global Economic Impact of the Year 2000 Software Problem," Software Productivity Research, Inc., January 23, 1997, page 40.

³*Ibid.*, page 79

⁴Warren S. Reid and Steven Brower, "Beyond Awareness: Ten Management and Ten Legal Pitfalls Regarding the Year 2000 Computer Problem That You May Not Have Considered, Yet!," <http://www.year2000.com/archive/beyond.html>, page 12.

⁵Gary H. Anthes, "Killer Apps," *Computerworld*, July 7, 1997, page 73.

⁶*Ibid.*, page 73.

⁷Thomas Hoffman, "Year 2000 Problem Comes Bundled with Legacy of Potential Litigation," *Computerworld*, October 14, 1996, page 88.

⁸Jeff Jinnett, "Legal Issues Concerning the Year 2000 Millennium Bug," <http://www.year2000.com/archive/legalissues.html>, pages 13-14.

⁹Warren S. Reid and Steven Brower, "Beyond Awareness: Ten Management and Ten Legal Pitfalls Regarding the Year 2000 Computer Problem That You May Not Have Considered, Yet!," <http://www.year2000.com/archive/beyond.html>, page 12.

¹⁰Ibid, page 13.

¹¹Ibid, page 16.

¹²Ibid, page 20.

¹³Ibid, page 45.



CHAPTER 10: LIFE INSURANCE ON YOUR BUSINESS?

¹"Business Bulletin," *The Wall Street Journal*, February 20, 1997, page 1.

²Thomas Hoffman, "CIOs Wary of Year 2000 Insurance," *Computerworld*, February 3, 1997, page 1.

³Robert L. Scheier, "Cashing in on Year 2000," *Computerworld*, July 14, 1997, page 59.

⁴Ibid.



CHAPTER 11: HUMAN RESOURCE ISSUES

¹David Weldon, "Deals of the Century," *Computerworld*, December 9, 1996, page 98.

²Julia King, "Staring Down the Deadline," *Computerworld*, June 2, 1997, page 69.

³"Year 2000 Scoreboard: Y2k a Yawner," *Computerworld*, March 9, 1998, page 80.

⁴Michael J Mandel and others, "Zap! How the Year 2000 Bug Will Hurt the Economy," *Business Week*, March 2, 1998, page 93.

⁵Ibid, page 97.

⁶"Year 2000 Scoreboard: Offshore Programmers: Low Cost for Long?" *Computerworld*, July 14, 1997, page 61.

⁷Patrick Canniff, "Tell Me Again Why I Need a Year 2000 Project Office," found at <http://www.year2000.com/archive/office.html>.

⁸Thomas Hoffman, "Few Contingency Plans Reported," *Computerworld*, June 9, 1997, page 6.



CHAPTER 12: PROJECT MANAGEMENT ISSUES

¹Julia King, "They Saw, They Planned, They Won," *Computerworld*, June 9, 1997, page 67.

²Patrick Dryden, "Millennium Puts Big Projects on Back Burner," *Computerworld*, June 1, 1998, page 1.

- ³Robert L. Scheier, "Many Year 2000 Projects Still at Starting Gate," *Computerworld*, April 14, 1997, page 24.
- ⁴William Farrell and Charles E. Phillips, "Year 2000 Watch: Survey at Year 2000 Conference," Morgan Stanley U.S. Investment Research, April 25, 1997, page 4.
- ⁵"Briefs: The Fortune 500 and the Year 2000," *Computerworld*, March 9, 1998, page 37.
- ⁶Robert L. Scheier, "Year 2000 Scoreboard," *Computerworld*, December 9, 1996, page 85.
- ⁷Charles E. Phillips and William E. Farrell, "Technology: Enterprise Software; Y2k Watch: Digital Plagues Oozes Across Planet," Morgan Stanley U.S. Investment Research, March 26, 1997, page 6.
- ⁸"Briefs: You Wait, You Pay," *Computerworld*, April 13, 1998, page 47.
- ⁹Capers Jones, "The Global Economic Impact of the Year 2000 Software Problem," Software Productivity Research, Inc., January 23, 1997, page 41.
- ¹⁰*Ibid*, page 18.
- ¹¹*Ibid*, pages 10, 11, and 14.
- ¹²*Ibid*, page 31.



CHAPTER 13: ENTERPRISE ISSUES

- ¹"Bureau of National Affairs Daily Report for Executives," November 20, 1996, A16.
- ²Capers Jones, "The Global Economic Impact of the Year 2000 Software Problem," Software Productivity Research, Inc., January 23, 1997, page 41.
- ³Scott Kirsner, "Millennial Blues: Preparing for Year 2000," *Boston.com*, <http://www.boston.com/business/emerging/year2000.htm>.
- ⁴Capers Jones, "The Global Economic Impact of the Year 2000 Software Problem," Software Productivity Research, Inc., January 23, 1997, page 42.
- ⁵*Ibid*.
- ⁶*Ibid*.
- ⁷*Ibid*.
- ⁸Bob Violino, "The Euro: Are You Ready?" *Information Week*, January 26, 1998, www.techweb.com.
- ⁹Nancy A. Cohen, "The Euro: Bigger than the Year 2000 Issue?" AICPA IT Section Tech Alert, February 1998.
- ¹⁰*Computerworld*, *Computerwoche*, and IDG News Service, "Your Other Millennium Problem," *Computerworld*, May 26, 1997, page 70.
- ¹¹"Year 2000 Scoreboard: On the Bright Side," *Computerworld*, March 9, 1998, page 80.
- ¹²David J. Lynch, "Denial, Inaction Threaten USA," *USA Today*, April 13, 1998, page 3B.

¹³James Cox, "Already Hard-Hit Region Virtually Unprepared," *USA Today*, April 13, 1998, page 3B.

¹⁴"Around the World: 2000 Bug Doesn't Worry Russian Atomic Agency," *The Dallas Morning News*, June 20, 1998, page 16A.

SECTION III: YEAR 2000 SOLUTIONS



CHAPTER 14: PROJECT INITIATION

¹"IT Acquisitions Year 2000 Compliance Update," U.S. Geological Survey, found at <http://ticindy.er.usgs.gov/ticcal/adp.html>, last modified September 30, 1996.



CHAPTER 15: SYSTEMS ASSESSMENT

¹Patrick Dryden, "Cover Your Assets," *Computerworld*, May 26, 1997, page 6.

²See, for example, http://www.isd.mcg.edu/PAGES/ISD/Y2k/let_txt.htm and <http://www.y2kexperts.com>.

³Randy Weston, "Oracle App Users Must Migrate," *Computerworld*, April 14, 1997, page 1.

⁴Ed Yourdon, "Year 2000 Also Involves Noncritical Systems," *Computerworld*, April 20, 1998, page 72.

⁵"Briefs: The Fortune 500 and the Year 2000," *Computerworld*, March 9, 1998, page 37.

⁶Gary H. Anthes, "When Disaster Strikes," *Computerworld*, January 19, 1998, page 80.

⁷Capers Jones, "Contingency Planning for the Year 2000 Software Problem," December 5, 1997, found at <http://www.erols.com/steve451/Caper.htm>.



CHAPTER 16: SYSTEM RENOVATION

¹Ed Yourdon, "Year 2000 Advice: Plan Now for Testing," *Computerworld*, March 16, 1998, page 69.

²Gary H. Anthes, "When Disaster Strikes," *Computerworld*, January 19, 1998, page 80.



CHAPTER 18: DATE MODIFICATION OPTIONS

¹John Raz, "City of Dallas Year 2000 Project," speech given to the Dallas-Fort Worth Data Administration Management Association Year 2000 Users Group, June 19, 1998.

²Tim Ouellete, "Insurer 'Lies' to Avoid Year 2000 Costs," *Computerworld*, July 28, 1997, page 1.



CHAPTER 19: Y2K CONSULTANTS

- ¹John M. Covalleski, "Some Firms Shy Away from Y2k Engagements," *Accounting Today*, June 8–21, 1998, page 4.
- ²Charles E. Phillips and William Farrell, "Y2K Watch: Digital Plague Oozes Across Planet," Morgan Stanley U.S. Investment Research Memorandum on Technology: Enterprise Software, March 26, 1997, page 12.
- ³Ibid.
- ⁴Ibid.
- ⁵Ibid.
- ⁶Ibid.
- ⁷"What Are You Waiting For? Start Preparing For Your Year 2000 Software Conversion Today," The Information Technology Association of America, 1996, page 7, <http://www.ita.org/yr2000bg.htm>.
- ⁸Ibid.



CHAPTER 20: TOOLS

- ¹"Year 2000 Scoreboard," *Computerworld*, March 9, 1998, page 80.
- ²Amy Malloy, "Beat the Clock!," *Computerworld*, January 6, 1997, page 67.
- ³C. Lawrence Meador, "Technology Overview: Solving the Year 2000 Problem," *Information Week*, February 5, 1996, <http://www.techweb.cmp.com/iw/565/65olr2.htm>.
- ⁴Cathleen A. Gagne, "This Isn't a Sprint . . . It's a Year 2000 Marathon," *Computerworld*, March 31, 1997, page 85.
- ⁵Ibid., page 85.
- ⁶Amy Malloy, "Beat the Clock!," *Computerworld*, January 6, 1997, page 67.
- ⁷Thomas Hoffman, "Client/Server Users Must Push for Year 2000 Tools," *Computerworld*, June 23, 1997, page 43.
- ⁸Amy Malloy, "Beat the Clock!," *Computerworld*, January 6, 1997, page 67.
- ⁹Ibid.
- ¹⁰Anna Fortig, "The Path To Year 2000 Is a Tool Road," *The Year 2000 Solutions Series*, *Computerworld Custom Publications*, June 2, 1997, page 16.
- ¹¹Robert L. Scheier and Thomas Hoffman, "Cashing in on Year 2000," *Computerworld*, July 14, 1997, page 59.
- ¹²Cathleen A. Gagne, "This Isn't a Sprint . . . It's a Year 2000 Marathon," *Computerworld*, March 31, 1997, page 85.
- ¹³Amy Malloy, "Beat the Clock!," *Computerworld*, January 6, 1997, page 67.

- ¹⁴Thomas Hoffman, "Client/Server Users Must Push For Year 2000 Tools," *Computerworld*, June 23, 1997, page 43.
- ¹⁵Amy Malloy, "Beat the Clock!," *Computerworld*, January 6, 1997, page 67.
- ¹⁶_____ "Tools, etc." *Computerworld*, January 19, 1998, page 89.
- ¹⁷Cathleen A. Gagne, "This Isn't a Sprint . . . It's a Year 2000 Marathon," *Computerworld*, March 31, 1997, page 85.
- ¹⁸Amy Malloy, "Beat the Clock!," *Computerworld*, January 6, 1997, page 67.
- ¹⁹Anna Fortig, "The Path To Year 2000 Is a Tool Road," *The Year 2000 Solutions Series*, *Computerworld Custom Publications*, June 2, 1997, page 16.
- ²⁰Amy Malloy, "Beat the Clock!," *Computerworld*, January 6, 1997, page 67.
- ²¹Capers Jones, "The Global Economic Impact of the Year 2000 Software Problem," Software Productivity Research, Inc., January 23, 1997, page 49.
- ²²Anna Fortig, "The Path To Year 2000 Is a Tool Road," *The Year 2000 Solutions Series*, *Computerworld Custom Publications*, June 2, 1997, page 16.

GLOSSARY

- Ada.** A programming language that was popular in the past with the government and some businesses.
- AS400.** A minicomputer produced by IBM.
- Assembler.** A second-generation programming language that some businesses used in the sixties and seventies. It is not very English-like and works directly with registers and machine instructions.
- BIOS (*basic or binary input output system*).** The part of the computer that executes the initial instructions when the computer is turned on.
- bridge.** A type of program that is built between two systems with different file formats. For example, if system one is converted to a four-digit year and system two remains a two-digit year, a bridge program must be written to translate the data between the two formats.
- C and C++.** Two programming languages. C++ is popular today and is considered an object-oriented language. It is used to program client/server and other business applications.
- change control.** The process of managing multiple copies of programs, including test versions and production versions. *See also version control.*
- CIO (*chief information officer*).** The head of an information systems department.
- client/server.** A type of system architecture where a server (a large computer) is connected to many workstations and the processing of data is shared between the server and the workstations.
- CMOS (*complementary metal-oxide semiconductor*).** A basic part of the computer that holds date and other startup information. It is battery operated when the computer is turned off. In AT-compatible computers, the CMOS contains a place to record century.
- COBOL (*COmmon Business-Oriented Language*).** A third-generation, English-like programming language, very popular through the eighties with businesses, and used primarily to program business applications.
- contingency plan.** An important part of a y2k plan. If the original plan fails, the contingency plan contains actions to execute to make up for the failure.
- crisis plan.** Part of a y2k plan that handles problems in 2000 that were overlooked or not completed. The crisis plan remains in effect until systems become stabilized.
- date compression.** One of the date conversion options. The four-digit year is compressed into a two-digit field.
- date expansion.** One of the date conversion options. The two-digit year is expanded into a four-digit field.

denial. What some executives experience when they first hear about y2k problems.

disaster recovery. A backup method should primary computers fail to process company data, whether the failure is from fire, theft, weather disaster or other catastrophe.

EDI (electronic data interchange). The process of exchanging business transactions in a standard format between two or more companies called trading partners.

embedded systems. These are computer chips that are everywhere! In our microwaves, VCRs, digital watches, car dashboard systems, thermostats, and washer/dryers, to name a few.

encapsulation. One of the date conversion options. Date data is set back twenty-eight years to avoid the y2k problem for another thirty years or so.

executive sponsor. The executive that has been assigned ultimate responsibility for a systems project. Also the project's champion and perhaps originator.

extranet. A part of a company's network that uses Internet technologies and is connected to outside suppliers and customers but is not generally accessible to the public at large.

firmware. The computer chip inside many electronics, appliances, and other devices that we depend upon today.

function points. A unit of measure of a program. Input, output, inquiries, logical files and interfaces are factors in considering how many function points a particular program has. It is used to express size of system or cost.

GO TO. A COBOL programming command that allowed the logic flow of the program to continue in another section of the program. The use of the GO TO statement was popular in the seventies before we realized how difficult those programs were to maintain and change.

hard code. A slang term used to describe programming code that contains fixed values. To change the values, the program must be changed and reimplemented. A system that was developed with a lot of hard coding is generally very inflexible and very costly to maintain.

IBM 40xx. A large mainframe computer produced by IBM. The last two digits are replaced with various numerals.

intranet. Refers to the use of Internet technologies inside an organization. An intranet does not have to be linked to the Internet.

LAN (local area network). A number of workstations and servers that are connected together via servers.

leap year. A year containing an extra day, February 29. 2000 is a leap year.

legacy systems. Older systems, commonly programmed in COBOL and executed on a mainframe computer.

lines of code. A measurement of a unit of a program. A system size can be expressed in lines of code, and costs to build systems are expressed per line of code. It is disputed exactly how long a line of code is, and it varies by language.

methodology. A standard set of steps to follow when conducting a systems project.

network. Computers that are connected together.

network operating system (NOS). An operating system that runs on a server. It keeps track of users on workstations attached to it, and it allocates shared resources such as printers and disks.

object code. The programming module that the machine actually executes. It is created by compiling source code.

online banking. The process of receiving bank transactions and paying bills online.

outsourcing. Contracting with another firm to perform a company function, such as payroll or a systems project.

PASCAL. A programming language that was popular in the past and used in business application systems.

personal digital assistant (PDA). A handheld device that performs such tasks as scheduling and contact management.

pivot point. In windowing, the value at which the century change is decided. For example, if 50 is the pivot point, all values below 50 will be assumed to be in century 20, and all values above 50 will be assumed to be in century 19.

production. The set of files, commands, and programs that constitute systems that a company is currently executing on the computer.

Programming language I (PL/I). A programming language used in business application systems that was popular in the past.

project assessment. An after-the-fact project review to learn what went right and what went wrong.

project manager. The person who leads a systems project.

real-time clock (RTC). The part of PC firmware that keeps the time. In AT-compatible PCs, time is kept here in two-digit year format.

recontamination. Where y2k-compliant systems become noncompliant again due to bad data or bad habits. This can happen with interfaces or in spreadsheets, especially.

regression test. A test performed to make sure that the system still works as it did before any changes were made.

remediation. The process of correcting systems for y2k compliance.

request for proposal (RFP). A document that lists requirements that a company wants in a system. It is designed to facilitate or begin conversations between a software vendor or consulting firm and a company who needs a new system.

server. A large computer (generally a souped-up workstation) that is connected to a number of workstations.

silver bullet. A magic solution, a quick fix, a panacea. Chances are there will not be one for y2k.

software license. A legal document which allows a company or individual to use one or more copies of a software program.

source code. The program instructions that the programmer writes.

source management. The process of managing multiple copies of programs. *See also* **version control**.

supply chain. Refers to the interrelationship between companies that sell and purchase products from/to each other. With y2k, these interdependencies could cause businesses to fail.

system test. A test done to a set of programs to ensure they work together and perform according to the documentation.

trading partner. A company that exchanges business transactions with other companies using EDI.

triage. This word is generally used in the medical profession to prioritize the wounded into three groups to facilitate saving as many lives as possible: a group that won't make it, a group that is well enough to get along without medical care for a while, and a group that requires medical care or they will die but if they get medical care they will have a good chance of living. The last group should get the first priority medical attention.

trust, but verify. You may hear this new buzz phrase as a way to say "accept the vendor's word that a system is compliant, but perform your own due diligence tests to be sure."

unit test. A test done to one program to validate that program changes work correctly.

value added reseller (VAR). An organization that sells software and provides value to the buyer through consulting services, training, or other service.

version control. The process of maintaining multiple versions of source programs and object programs. A new version is created each time a program change is implemented. There can be a production version, a test version for each project underway, and previous generations of programs that must be managed.

Wide Area Network (WAN). A communications network that covers a wide geographic area.

windowing. One of the date conversion options. It is accurate for only a 100-year range of dates and requires logic in the programs that says if a date is below, say 50, it will be in the century 00. If the date is 50 or above, then it will be in the 19 century.

y2k. Shorthand for "year 2000."

Y2k compliant. There is no accepted definition of what this means, but it is a way to say that systems should work past 1999. You should create your own wording when working with any vendor.

Y2k ready. There is no accepted definition of what this means, but it is a way to say that a particular item should work past 1999 as long as it is run on other components that are also y2k-ready. (reminds me of plug and play) You should create your own definition when working with vendors.

Year 2000 problem. When a computer interprets an *00* in the year field to be the year 1900.

ABOUT THE AUTHOR

Sandi Smith, CPA, MBA, CMA, CDP, is based in Dallas, Texas and offers technology consulting and training to businesses around the globe. Sandi speaks frequently around the country on various technology topics, such as y2k, top 10 technologies, and Internet strategies. She has authored two editions of AICPA's *Top 10 Technology Opportunities* books and numerous articles for *The CPA Letter*, *Journal of Accountancy*, *Management Accounting*, and other publications. Sandi has developed a computer-based training course, a practice aid, videos, television segments, and instructor-led courses on the y2k topic. She is a member of the AICPA's Information Technology Research Subcommittee and Strategic Planning Advisory Subcommittee. In addition, she may be the only CPA who has flown a single-engine airplane around the world. She enjoys receiving comments and questions at sandi@sandismith.com.

AICPA INFORMATION TECHNOLOGY MEMBERSHIP SECTION INFORMATION TECHNOLOGY FOR CPAs BY CPAs

Keeping up with information technology can be a full-time job. The AICPA Information Technology Membership Section makes it easy for you to stay current on the latest technology and its uses. We offer unbiased professional advice written for CPAs by CPAs that can help you make and save money. Membership in the Section is open to all AICPA members and qualifying non-CPAs at all levels of technology expertise. Just take a look at what the Information Technology Membership Section has to offer . . .

PUBLICATIONS

The AICPA Information Technology Membership Section publishes these documents to help accountants obtain an understanding of the technologies that face them today as well as down the road. In addition, a handsome binder is provided for you to keep your newsletters and alerts neatly and conveniently within reach.

Bi Monthly Newsletter—*Info Tech Update*. A 12-page newsletter filled with practical articles on dealing with technology.

Technology Alerts—One page alerts discussing “hot technology topics.” Topics have included: *Updating Microsoft's Windows 95 and Internet Explorer*, *What's Hot, What's Not: COMDEX 96*, *HPCs and Windows CE—Worth the Wait?*, *The 809 Area Code*. . . , *The year 2000* . . . , *Electronic Cookies* . . . , and more. Due to rapid changes in the information technology environment, alerts are issued as newsworthy issues arise.

DISCOUNT OFFERINGS

Past examples include a 35% discount on *Franklynn Peterson's CPA Computer Report*. In addition, Section members received special pricing on the purchase of the book *1997 Top 10 Technologies and Their Impact on CPAs* (1997), as well as a 35% discount (\$19.50 for IT Section members) on the purchase of the book *EDI Control, Management and Audit Issues* (1995) No. 043004.

Section members automatically receive *The CPA Software News* as part of their membership benefit package.

If you're still not sure what membership in the Information Technology Membership Section can do for you, call the AICPA and we'll be glad to discuss the Section with you! Call or fax:

Andrew R. Gioseffi, CPA
Information Technology Membership Section
Phone: (212) 596-6211
FAX: (212) 596-6025
Email: infotech@aicpa.org

If you want to keep up with technology, join the Information Technology Membership Section today! Don't miss out on another newsletter, publication or discount program.

MEMBERSHIP APPLICATION FORM

AICPA Information Technology Membership Section

Please enroll me as a member in the AICPA Information Technology Membership Section. I understand that the \$100.00 (\$165.00*) dues covers all membership benefits.

I am returning this form along with my credit card number or check payable to the AICPA.

Name: _____

Firm: _____

Address: _____

City: _____ State: _____ Zip: _____

Member Number: _____

Business Telephone: _____

☐ Credit card # _____ VISA or M/C EXP DATE _____

Signature: _____

Fax Number: _____

Email address: _____

Mail or Fax to AICPA Member Satisfaction Team:

Member Satisfaction Team
American Institute of CPAs
PO Box 2206
Jersey City, NJ 07303-9946
Fax: (800) 362-5066

Please note that the dues will be prorated, based on when you enroll.

*Non-CPA Section Associate membership rate.

READER’S RESPONSE TO *Solving the Year 2000 Dilemma*

Your assessment of this book will help to ensure that future publications in this series will be of value to readers. Please complete this questionnaire and mail or fax it to **Murray B., Schwartzberg, Specialized Publications & Subsidiary Rights, AICPA, Harborside Financial Center, 201 Plaza Three, Jersey City, NJ 07311-3881; Fax: (201) 938-3780.**

Thank you for your assistance.

1. How knowledgeable were you about this topic before you read this publication?

1	2	3	4	5
No knowledge			Very knowledgeable	

2. How useful is this publication to your line of work?

1	2	3	4	5
No knowledge			Very knowledgeable	

What is your current job title? _____

3. What did you like most about this publication?

4. What did you like least about this publication?

5. Did the technical level of this publication meet your needs?

1	2	3	4	5
No knowledge			Very knowledgeable	

6. Did this publication live up to your expectations? Yes _____ No _____

7. Where did you learn about this publication?

CPA Letter _____

AICPA Website _____

Journal of Accountancy _____

InfoTech Update Newsletter _____

AICPA Conferences _____

Other (please explain) _____

Additional comments and suggestions _____

Name and address _____

Email address _____

Thank you for taking the time to complete this reader response survey!

Capitalize on New and Profitable Opportunities With AICPA's Information Technology Resources

The CPA's Guide to the Internet

By John Graves, CPA and Kim Hill Torrence

An all-encompassing guide to harnessing the power of the Internet for assisting and enhancing your business. This guide, updated for 1998, shows you how to tap into the profitable opportunities of the Internet: learn how to get connected and begin navigating the Net, see how to create an Internet site, learn about services CPAs can offer on the Net. Also included with the Guide is Kentis Internet Toolkit, a CD-ROM comprised of several software applications including Microsoft Internet Explorer and Microsoft NetMeeting, in addition to multimedia videos on how to use these and other Internet applications.

No. 093009KC \$44.00 / \$55.00 Nonmembers

NEW Edition! The CPA's Guide to Web Commerce

By John Graves, CPA and Jacqueline Justice, Ph.D.

From planning to implementation, this guide prepares you to meet the future of commerce on the World Wide Web. Learn how a Web site can enhance marketing efforts, either as a point of sale or in another marketing function, depending upon the type of product or service offered. In addition to the Guide, you will receive a CD-ROM containing software to construct a Web site, along with instructional videos that show how to complete the task.

No. 093011KC \$39.95 / \$47.95 Nonmembers

NEW Edition! The CPA's Guide to Intranets

By John Graves, CPA and Jacqueline Justice, Ph.D.

Take the mystery out of Intranets and learn how to maximize the benefits of these special private networks. Learn how an Intranet site can cut costs and enhance corporate administration. In addition to the Guide, you will receive a CD-ROM containing software to construct an Intranet site, along with instructional videos that show how to complete the task.

No. 093012KC \$39.95 / \$47.95 Nonmembers

NEW Edition! The CPA's Guide to Information Security

By John Graves, CPA and Kimberly S. Hill

The tools and techniques you need to establish, monitor, and maintain information security. This guide tackles the issue from the CPA's perspective, offering a thorough analysis of the problem as well as actions to take right now for better information security.

No. 093010KC \$39.95 / \$47.95 Nonmembers

NEW! The CPA's Guide to Managing Workflow with Technology

By John Graves, CPA, Kimberly S. Hill and Lisa F. Allen, MA

This Guide gives CPAs the skills and background they need to select workflow software, evaluate workflow issues in their own organizations and those of clients and manage work flow projects in both small businesses and larger organizations. Worksheets and case studies are included to provide CPAs with real-world examples of how workflow technology can

increase the efficiency and quality of organizational tasks and services. Also included, the *Business Process Reengineering Toolkit* is a valuable CD-ROM that provides a wealth of supplemental materials to support CPAs in various types of BPR initiatives, including: Electronic Data Interchange (EDI); Document Image processing (DIP); and Workflow Technologies.

No. 090436KC \$39.95 / \$49.95 Nonmembers

NEW! The CPA's Guide to Communications Technologies

By John Graves, CPA; Marge Cataldo, M.A. and Kimberly S. Hill

By supplementing the descriptions of various communications technologies with case studies and worksheets, *The CPA's Guide to Communications Technologies* offers a rich overview of this exciting new technology and a self-guided planning tool in one easy text. Also included, the *CommTech Tools* CD-ROM helps CPAs apply the concepts described in the manual to real-world business situations.

No. 090435KC \$39.95 / \$49.95 Nonmembers

NEW! The CPA's Guide to Document Image Processing

By John Graves, CPA, Jacqueline Justice, Ph.D. and Marge Cataldo, M.A.

Receive a wealth of how-to tools that support the efforts of CPAs as they apply this important technology to their organization and the organizations of their clients. Worksheets and case studies are included in the text and can be used for individual project planning or by entire project teams. Also included, the *Business Process Reengineering Toolkit* is a valuable CD-ROM that provides a wealth of supplemental materials to support CPAs in various types of BPR initiatives, including Electronic Data Interchange (EDI); Document Image processing (DIP); and Workflow Technologies.

No. 090434KC \$39.95 / \$49.95 Nonmembers

Survival Guide for Road Warriors

By Daniel S. Coolidge, Esq. and J. Michael Jimmerson, Esq.

This non-computerese book tells you everything you need to know about creating a successful "office on the go," including the use of Fax, E-mail, alternate communication channels, zipping and unzipping files, infrared ports, and other technologies that are indispensable to a true "road warrior".

No. 090426KC \$22.95 / \$28.95 Nonmembers

Creating a Virtual Office: Ten Case Studies for CPA Firms

By Anita Dennis, issued by the AICPA Management of an Accounting Practice Committee

Learn about establishing a successful virtual office through real-world case studies which cover: Traditional firms in unconventional settings; Using the virtual office to create a niche practice; A pension benefits boutique; How long-distance employees get a marketing edge; and much more!

No. 090426KC \$22.95 / \$28.95 Nonmembers

To Order, Call the AICPA at 1-888-777-7077

