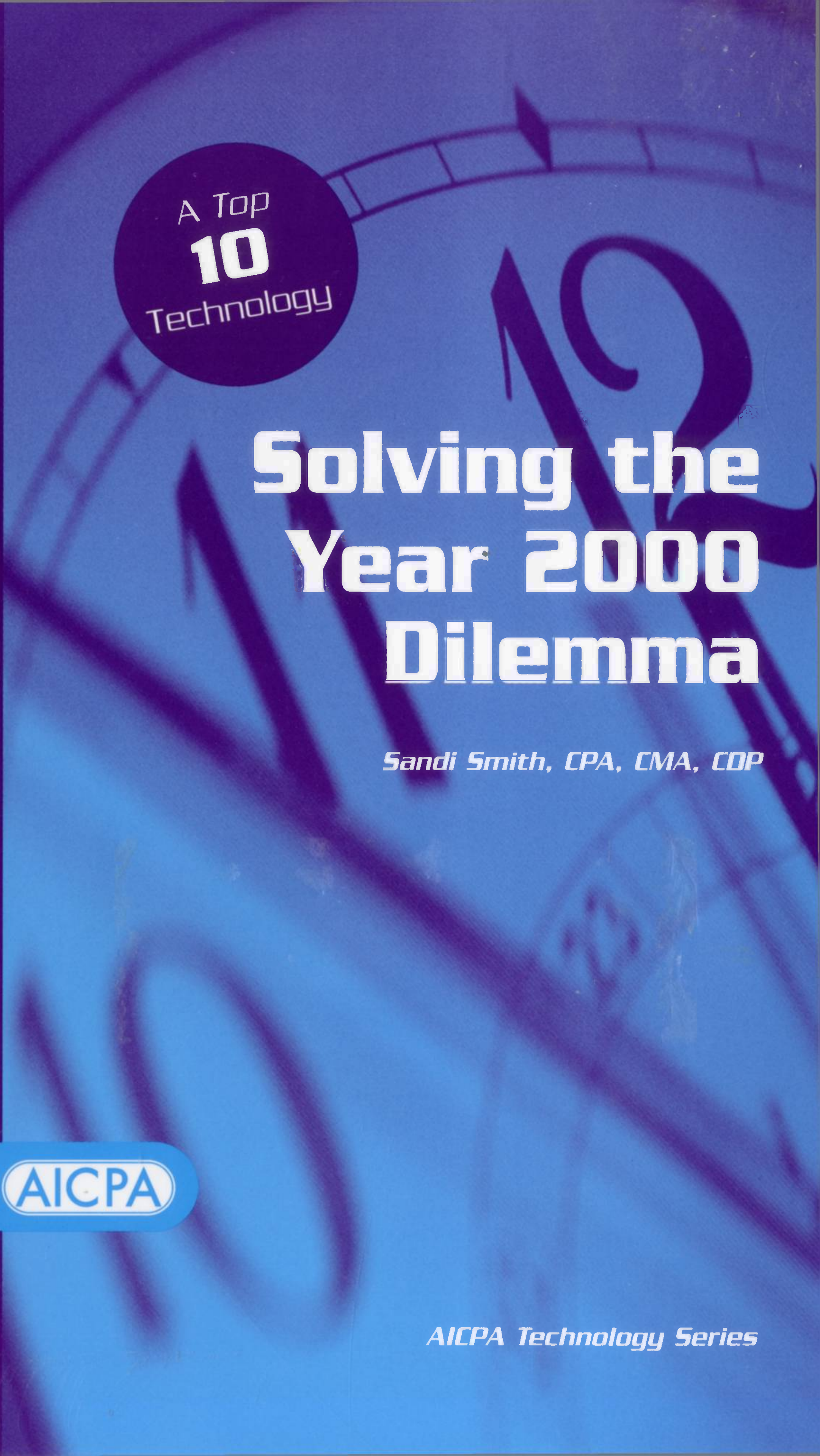
1997

Solving the year 2000 dilemma

Sandi Smith



A Top
10
Technology

Solving the Year 2000 Dilemma

Sandi Smith, CPA, CMA, CDP

AICPA Technology Series

American Institute of Certified Public Accountants

AICPA

Solving the Year 2000 Dilemma

AICPA

A Top 10 Technology

Solving the Year 2000 Dilemma

Sandi Smith, CPA, CMA, CDP

American Institute of Certified Public Accountants

AICPA

AICPA Technology Series

Notice to Readers

Solving the Year 2000 Dilemma does not represent an official position of the American Institute of Certified Public Accountants, and it is distributed with the understanding that the author and publisher are not rendering legal, accounting, or other professional services in this publication. If legal advice is required, the services of a competent professional should be sought.

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

ISBN 0-87051-196-3

Dedication

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

Table of Contents

Section I

The Year 2000 Dilemma

WHEN THE YEAR 2000 COMES, YOUR COMPUTERS WILL THINK IT'S THE YEAR "00" AND CAUSE MAJOR PROBLEMS.
www.unitedmedia.com
S. Adams
THE DOGBERT CONSULTING COMPANY CAN FIX THE PROBLEM FOR ONLY TEN MILLION DOLLARS. OUR WORK IS GUARANTEED FOR ONE FULL YEAR, STARTING TODAY.
© 1996 United Feature Syndicate, Inc.(NYC)
9/17/96
BUT WHY WOULD I CARE? THE YEAR "00" IS BEFORE I'M BORN.
AMAZING... YOU'D ACTUALLY HAVE TO BE SMARTER TO DO SOMETHING STUPID.

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—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, "Y2k 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, "Y2k 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 dilemma and its major wrinkles and problems. Chapter 1 will describe the scope of the problem and will define it. Chapter 2 will take a look at how each of us will be affected personally by the dilemma. In chapter 3, I'll take a business perspective, discussing the impact on hardware, software, and systems you might not normally think of. Chapter 4 will help us look outside our organizations for y2k problems; chapter 5 will warn us about the leap year day in February 2000; and chapter 6 will caution us about "silver bullets."

CHAPTER 1:
IT'S BIGGER THAN A BREADBOX

The problem 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 year 2000 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 problems are staggering. The Gartner Group's $300–$600 billion estimate 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 together, 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.[3]

Event	*Estimated Cost*
World War II	$4,200B
Year 2000	$600B
Vietnam conflict	$500B
Kobe earthquake	$100B
Los Angeles earthquake	$40B

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 y2k. 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.[4]

The estimated y2k price tags for large companies are now appearing in the press. Chase Manhattan Bank expects to spend up to $250 million. Electronic Data Systems Corp. plans to spend $144 million.[5] The federal government could wind up with a $30 billion tab.

Some say the $600 billion figure 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 costs[6]

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.[7]

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. Our 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 will 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."[8]

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 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 because it thought they had completed their sentences.[9] 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, where 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.[10]

Exactly how did these events happen? 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 (that is, we programmed fixed, unchanging values) tables inside the program to convert "06" to "June," for example, and "83" 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," *18*93, that is. 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 '60s, would make it until 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 deadline.

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 of 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-eighties, 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 will be a question pondered by historians in the technology industry for decades to come.

Although small businesses will not have the hefty price tag of the Fortune 500s, they will not be able to ignore the dilemma entirely. Just about every business will see some of the problems the dilemma creates.

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 where I am a customer. First, my credit card companies.

CHECK YOUR WALLET

If you think that the y2k dilemma hasn't affected you personally yet, check inside your wallet. Pull out your credit cards and check their expiration dates. As of late 1997, most of your cards will probably show an expiration date of no later than December 1999.

There are three hurdles regarding the use of credit cards with an expiration date past 1999:

1. The credit card company must be compliant. These include the giants Visa USA, Inc. and MasterCard International.
2. The member banks must be compliant.
3. The merchants' card readers must be compliant.

The last item will cause the biggest problem. Many merchants are unaware that there are problems with their equipment. The problem hasn't appeared yet in large numbers because the major companies, such as MasterCard International, Inc. and Visa USA, Inc., have requested that their member banks not produce cards with the "00" expiration date until systems are corrected.[1]

But there have been a few isolated problems already. First Visa USA, Inc. recalled a few thousand cards with "00" expiration dates earlier this year. And 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.[2]

Losing a sale because of a bad card reader can easily be prevented. CPAs who work for companies or who have clients that take credit cards can advise them to check their card readers immediately. Equipment that is not found to be compliant must either be replaced or updated with a new software version. The credit card equipment vendor can be called to find out the details of becoming compliant.

The oil and gas companies seem ahead on this issue. As an example, Fina cards and Exxon cards that expire in 2000 were in circulation as of mid-1997. A spokesperson for Discover said they will begin issuing credit cards that expire in 2000 before the end of 1997. In August 1997, the company was in the process of notifying its merchants. A great opportunity exists for CPAs to give their employers and clients a "heads-up" warning about their credit card processing equipment.

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 4 years has had to cut back to 3-year expiration dates to work around the problem.[3]

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 $250 million on its y2k effort to ensure that its $trillion-a-day transaction level is not threatened. Although $250 million sounds like a lot, Chase spends $1.8

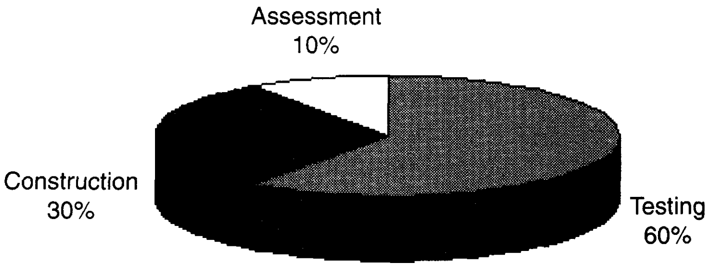
billion a year on technology, therefore its y2k project will gobble up only 12–13 percent of its total technology budget.[4] The Chase company expects to fund the project from its existing technology budget.

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.[5]

The project will be handled in three phases:

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 ensure that code works together as planned.[6]

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 this year for employees to peruse to learn which software products are compliant and which are not.[7]

MORE BREAD—INVESTMENTS

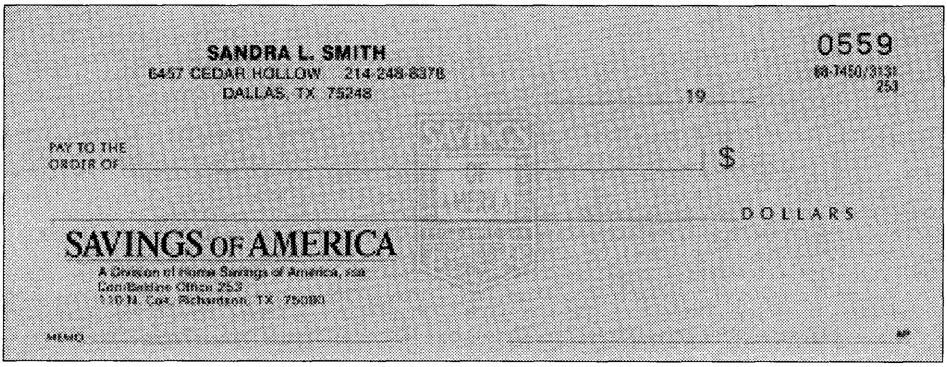
I called T. Rowe Price, a mutual funds company. Rowena Itchon, a spokesperson for the company, said its systems are new, and the company would not have a big problem with the y2k.

The securities industry, as a whole, is expected to spend $4 billion on the problem.[8] Companies in this group include stock brokers, 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]

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 it say "DATE__________" or does it say "________19__" on the date line?

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

________19____

PAY TO THE
ORDER OF ________________ $

________________ DOLLARS

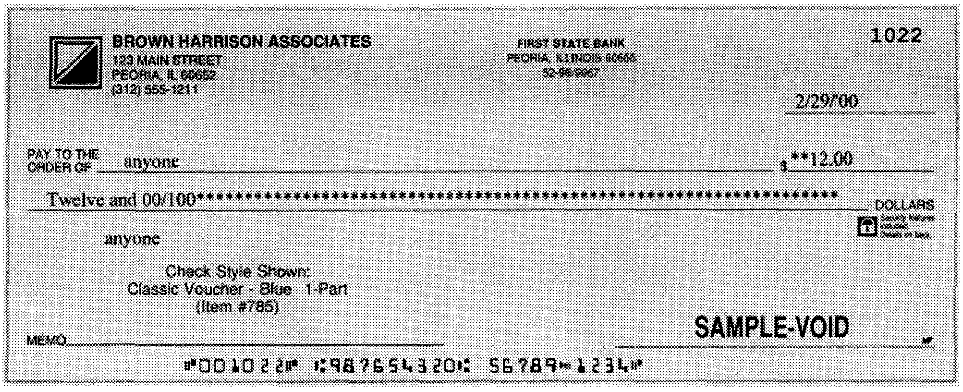
SAVINGS OF AMERICA
A Division of Home Savings of America, FSB
ConBanking Office 253
110 N. Coit, Richardson, TX 75080

MEMO ________________

Most forms have been changed already, and I realize that this is a trivial issue because you 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 quote 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.

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
52-96/9967

1022

2/29/'00

PAY TO THE ORDER OF anyone $ **12.00

Twelve and 00/100*** DOLLARS

Security features included. Details on back.

anyone

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

MEMO

SAMPLE-VOID

⑈001022⑈ ⑆987654320⑆ 56789⑈1234⑈

The best solution is to bank online, where check forms are unnecessary.

One consulting firm is advocating obtaining hard copies of statements of all of the investment and bank accounts that an individual owns.[10] Although this is a good idea in general and especially in case of a fire, Internal Revenue Service (IRS) audit, or other disaster, it is probably overkill for y2k purposes.

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, where systems are being identified and prioritized as to criticality.

I asked him what the worst-case scenario could be (but probably wouldn't be, because it would be fixed in time for y2k). 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 '60s and '70s 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 if 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 is 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 two y2k initiatives:

1. Information technology (IT), which consists of the mainframe and application software. This encompasses systems such as billing and payroll.
2. 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 complete by the end of 1998.

The corporate initiative has recently been started and an assessment is underway. I asked Mr. Keating what the worst-case scenario could be (but won't occur because we know he is on top of it), and he 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 him if he thought there might be any safety issues. He said no. He said that many of the control systems were PC-driven.

A recent 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. We hope the engineer finds what he is looking for.

PHONES

Telephone companies must not only fix their billing and accounting systems, but 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.[11]

CARS AND PLANES

I called Honda to find out if my car would not start on January 1, 2000 as *Newsweek* suggested.[12] I'm glad to report that the car's computers keep track of the odometer reading instead of the date. So chances are very high that your car will start on January 1. Unless, of course, you have pulled an "all-nighter" at a certain once-in-a-century party, and you left your lights on when you pulled up to park the night before.

Aircraft manufacturers such as Boeing rely on hours of operation instead of dates, so the y2k limitation does not apply to its airframes.[13]

Although airplane engines may not have a problem, navigation systems could. They will not have a y2k problem. 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.[14,15] GPS units are popular with pilots, airlines, rental car companies, campers, sailors, and hobbyists.

VCRs and Other Appliances

An easy way to tell if a household appliance could have a problem is to see if it holds a date in the first place. My VCR is on a 7-day, 24-hour cycle, so it is not smart enough to know what year it is. It won't care whether the y2k comes and goes.

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

Government Services

Most government agencies and many state and local agencies will not be prepared for y2k. The Gartner Group says 30 percent of the federal government's systems will not be compliant by the deadline. And the states are in worse shape. In 1996, a survey by the National Association of State Information Resource Executives disclosed that 40 percent of states have not started y2k projects.[16]

There are a few bright stars. The Social Security Administration started 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 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."[17] 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."[18]

If you have any business with the government, try to get it out of the way early. If you think there are excuses now, just wait until . . .

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. Last, 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 5 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 PCs will work just fine past 1999.

The problem with PCs and the y2k is one of the most minor issues we'll see in our technology experiences. In most cases, you can basically do nothing until the first time you turn your computer on in the next century. 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.

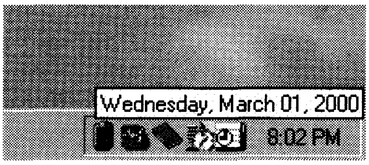
A Little Test

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, there is one warning. 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.

When I turned my laptop back on after 5 minutes, I checked the system date for a correct century display. It read January 1, 2000. It passed. If your computer passes this part of the test, you can then skip the next few steps and move to the software part of the test.

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



Wednesday, March 01, 2000

8:02 PM

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 if 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.

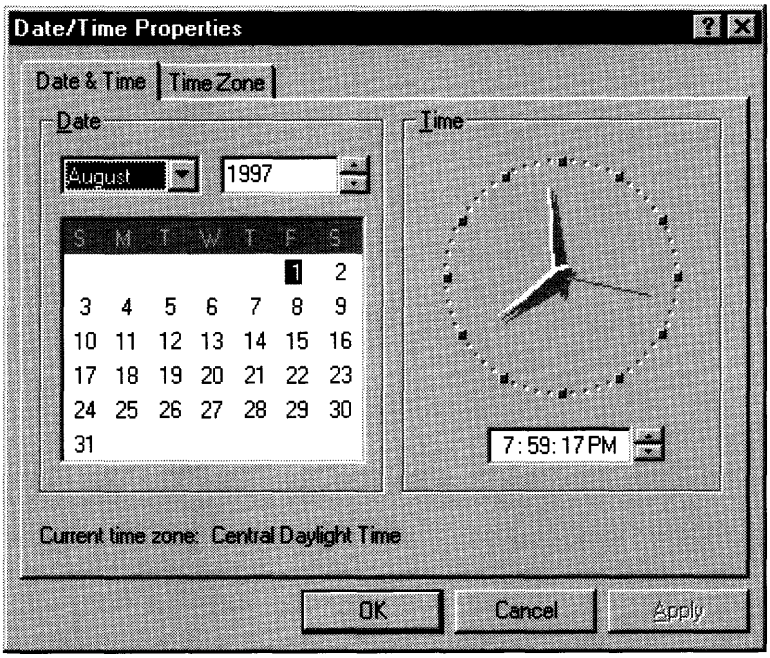
RESETTING THE DATE

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.

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.

FIGURE 3.2: WINDOWS 95 DATE/TIME PROPERTIES



Date: August 1997

S	M	T	W	T	F	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

Time: 7:59:17PM

Current time zone: Central Daylight Time

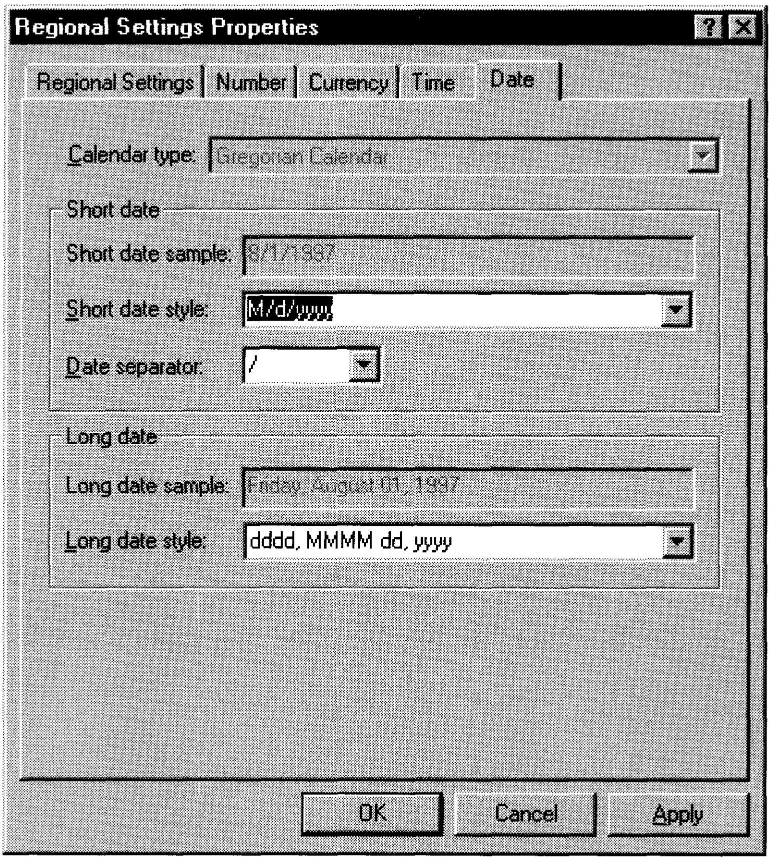
At the DOS (disk operating system) prompt on a DOS PC, you can type DATE and the system will return the current date and prompt you for the new date in MM-DD-YY format. You can then enter 01-01-00. You can also set the time in a similar fashion via the TIME command.

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.[2]

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 if needed, if the battery is running low, if a time zone change occurs, or if a similar issue occurs to render the date value incorrect.[3]

FIGURE 3.3: WINDOWS 95 DATE STYLES



Calendar type: Gregorian Calendar

Short date

Short date sample: 8/1/1997

Short date style: M/d/yyyy

Date separator: /

Long date

Long date sample: Friday, August 01, 1997

Long date style: dddd, MMMM dd, yyyy

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. The 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.[4] This is a type of error message display, albeit cryptic.

OPTIONS FOR PCS

There are many options for PCs. Chances are high you will not have to throw away your old PC!

1. If your PC is networked, it can get the date from the network and it will run just fine without any intervention. I recommend this method for large businesses—it seems like the most cost-effective approach in most cases.
2. Most operating systems will be fixed to compensate for any hardware glitches that remain in the BIOS.
   - Apple boasts that MacOS (Macintosh Operating System) has been compliant since its inception in 1984. It correctly recognizes years from 1920 to 2019.[5]
   - 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.[6]
   - Windows NT 4.0 is able to produce the correct date.[7]
   - Windows 95, DOS, Windows 3.1, and Windows for Workgroups currently do not automatically correct for the date. Once the date is changed by the PC user after the century change, the system will hold the date.
3. If you had to change the date but your system will hold the date, the following steps are optional. If you choose to bypass them, you will have to change the date manually. If you want the computer to roll over automatically, then you should do one of the following steps. If your PC will be running mission-critical applications just after the millennium and you cannot physically be there to monitor the jobs, this is advised. Also, if your system will not hold the date, you can try one of the following steps.
   a. If the PC's BIOS is able to 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 y2k.

   b. If your BIOS is not able to be modified, the problem can be corrected using a driver or adding a code early in the *config.sys* file for DOS applications.

   c. There are some companies producing utilities that will update the date. The RighTime Company offers products to test and correct the y2k problem on PCs. They can be found at http://www.rightime.com. Please check out any vendor's promises very thoroughly; I am listing this product for completeness and do not endorse any product listed.

4. If all of the above fails, you can simply set the time the first time you turn on your PC in the new century. This is what I recommend for small businesses; it is the most cost-effective approach.
5. If the PC fails to hold the date after trying all of the above, it's time to:
   - 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 y2k. 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. A few of the hardware vendors' Web pages are listed below for your convenience.

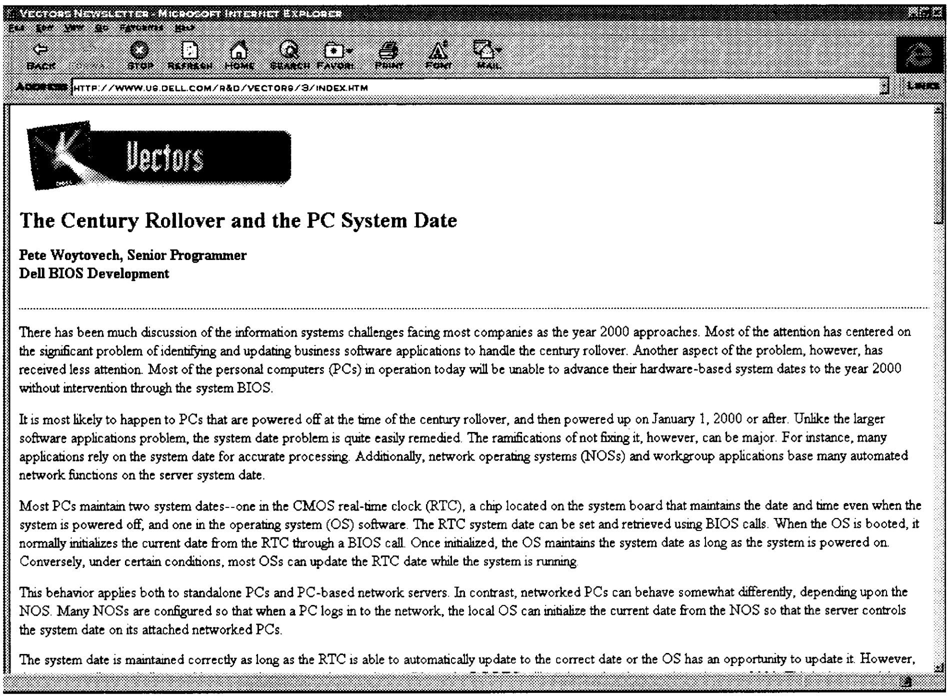
- Acer—http://www.acer.com/aac/support/articles/yr2000.htm.
- AST—http://www.ast.com/americas, then search for "year 2000." You'll get a press release and two technical bulletins that are pertinent.
- Compaq—http://www.compaq.com, enter "year 2000" in the search engine.

- Dell—http://www.us.dell.com/support/tech/rm/r&dnews/vectors/3/v3_cent.htm.
- Gateway—http://www.gw2k.com.
- Hewlett-Packard (HP)—http://www.hp.com/cposupport/personalcomputing/support_doc/bpv03013.html.
- IBM—http://www.ibm.com/IBM/year2000/.
- Micron—http://www.micronpc.com/support/faq/topissues/ts16394.html.

In addition, a few y2k Web sites list vendors' positions on particular PC models:

- The state of Washington, at http://www.wa.gov/dis/2000/survey/dt_hard/.
- The Computer Information Centre, at http://www.compinfo.co.uk/y2k/manufpos.htm.

FIGURE 3.4: DELL'S Y2K PAPER


VECTORS NEWSLETTER - MICROSOFT INTERNET EXPLORER
File Edit View Go Favorites Help
BACK
STOP
REFRESH
HOME
SEARCH
FAVORI
PRINT
FONT
MAIL
ADDRESS
HTTP://WWW.US.DELL.COM/R&D/VECTORS/3/INDEX.HTM
LINKS
Vectors

The Century Rollover and the PC System Date

Pete Woytovech, Senior Programmer
Dell BIOS Development

There has been much discussion of the information systems challenges facing most companies as the year 2000 approaches. Most of the attention has centered on the significant problem of identifying and updating business software applications to handle the century rollover. Another aspect of the problem, however, has received less attention. Most of the personal computers (PCs) in operation today will be unable to advance their hardware-based system dates to the year 2000 without intervention through the system BIOS.

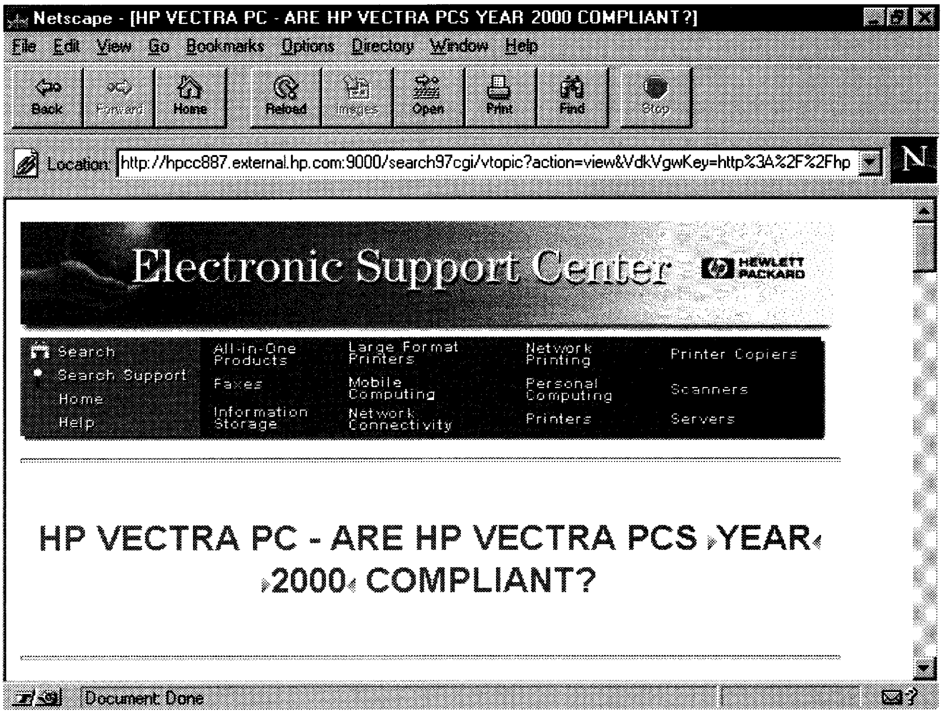
It is most likely to happen to PCs that are powered off at the time of the century rollover, and then powered up on January 1, 2000 or after. Unlike the larger software applications problem, the system date problem is quite easily remedied. The ramifications of not fixing it, however, can be major. For instance, many applications rely on the system date for accurate processing. Additionally, network operating systems (NOSs) and workgroup applications base many automated network functions on the server system date.

Most PCs maintain two system dates--one in the CMOS real-time clock (RTC), a chip located on the system board that maintains the date and time even when the system is powered off, and one in the operating system (OS) software. The RTC system date can be set and retrieved using BIOS calls. When the OS is booted, it normally initializes the current date from the RTC through a BIOS call. Once initialized, the OS maintains the system date as long as the system is powered on. Conversely, under certain conditions, most OSs can update the RTC date while the system is running.

This behavior applies both to standalone PCs and PC-based network servers. In contrast, networked PCs can behave somewhat differently, depending upon the NOS. Many NOSs are configured so that when a PC logs in to the network, the local OS can initialize the current date from the NOS so that the server controls the system date on its attached networked PCs.

The system date is maintained correctly as long as the RTC is able to automatically update to the correct date or the OS has an opportunity to update it. However,

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



Netscape - [HP VECTRA PC - ARE HP VECTRA PCS YEAR 2000 COMPLIANT?]

Location: http://hpcc887.external.hp.com:9000/search97cgi/vtopic?action=view&VdkVgwKey=http%3A%2F%2Fhp

HP VECTRA PC - ARE HP VECTRA PCS ›YEAR‹ ›2000‹ COMPLIANT?

Document: Done

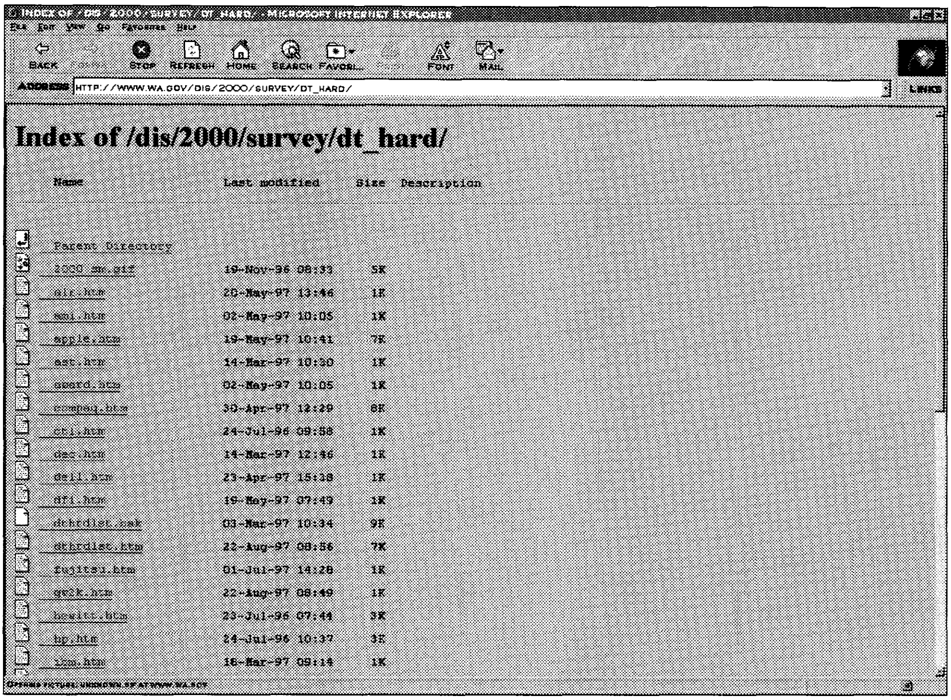
MISCELLANEOUS HARDWARE DEVICES

The y2k dilemma does not stop with PCs and laptops. 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.

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 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.

FIGURE 3.6: THE STATE OF WASHINGTON LISTS VENDORS' Y2K COMPLIANCE LETTERS


Index of /dis/2000/survey/dt_hard/ - Microsoft Internet Explorer
File Edit View Go Favorites Help
Back
Forward
Stop
Refresh
Home
Search
Favori...
Font
Mail
Address
Links

HTTP://WWW.WA.GOV/DIS/2000/SURVEY/DT_HARD/

Index of /dis/2000/survey/dt_hard/

```
Name                    Last modified     Size  Description

Parent Directory
2000_sm.gif             19-Nov-96 08:33     5K
air.htm                 20-May-97 13:46     1K
ami.htm                 02-May-97 10:05     1K
apple.htm               19-May-97 10:41     7K
ast.htm                 14-Mar-97 10:30     1K
award.htm               02-May-97 10:05     1K
compaq.htm              30-Apr-97 12:29     8K
cti.htm                 24-Jul-96 09:58     1K
dec.htm                 14-Mar-97 12:46     1K
dell.htm                23-Apr-97 15:38     1K
dfi.htm                 19-May-97 07:49     1K
dthrdlst.bak            03-Mar-97 10:34     9K
dthrdlst.htm            22-Aug-97 08:56     7K
fujitsu.htm             01-Jul-97 14:28     1K
gw2k.htm                22-Aug-97 08:49     1K
hewitt.htm              23-Jul-96 07:44     3K
hp.htm                  24-Jul-96 10:37     3K
ibm.htm                 18-Mar-97 09:14     1K
```

Opening picture: unknown.gif at www.wa.gov

Software packages that do not use dates extensively, such as word processing programs, graphics, and desktop publishing, will not be mentioned because the problems are generally nonexistent or very minor.

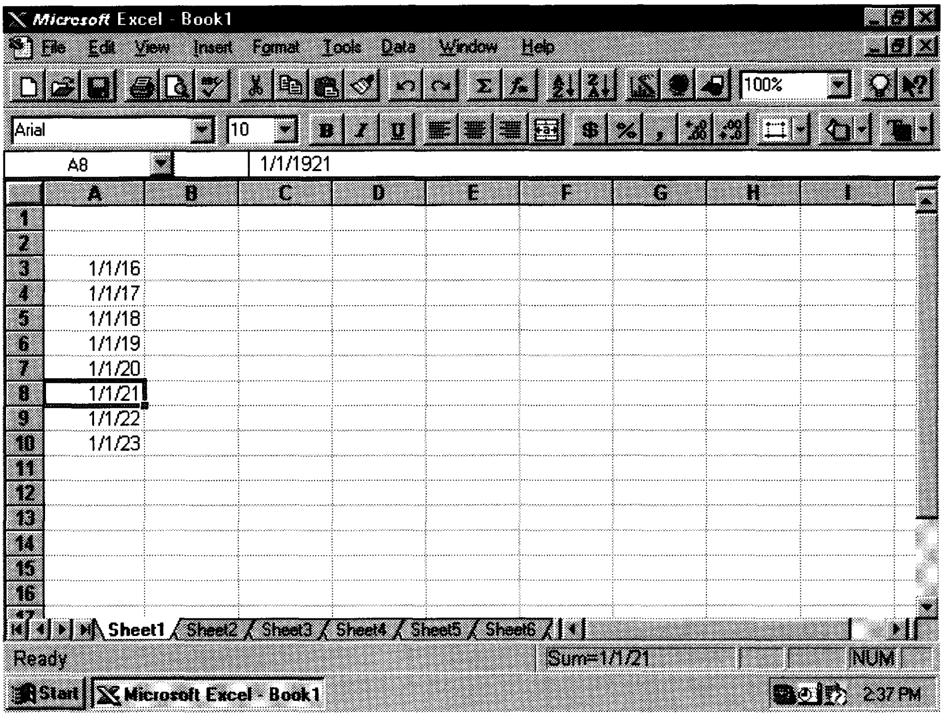
Spreadsheets

Microsoft Excel Version 7 for Windows 95 handles years in the twenty-first century with only minor intervention. 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.[8]

FIGURE 3.7: MICROSOFT EXCEL AND THE Y2K DILEMMA



A8 | 1/1/1921

	A
1	
2	
3	1/1/16
4	1/1/17
5	1/1/18
6	1/1/19
7	1/1/20
8	1/1/21
9	1/1/22
10	1/1/23
11	
12	
13	
14	
15	
16	

Ready | Sum=1/1/21

Corel Quattro Pro 8 (and also Corel Quattro Pro 7 for Windows NT) works similarly, but with different date range assumptions. 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.[9]

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 the 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.[10]

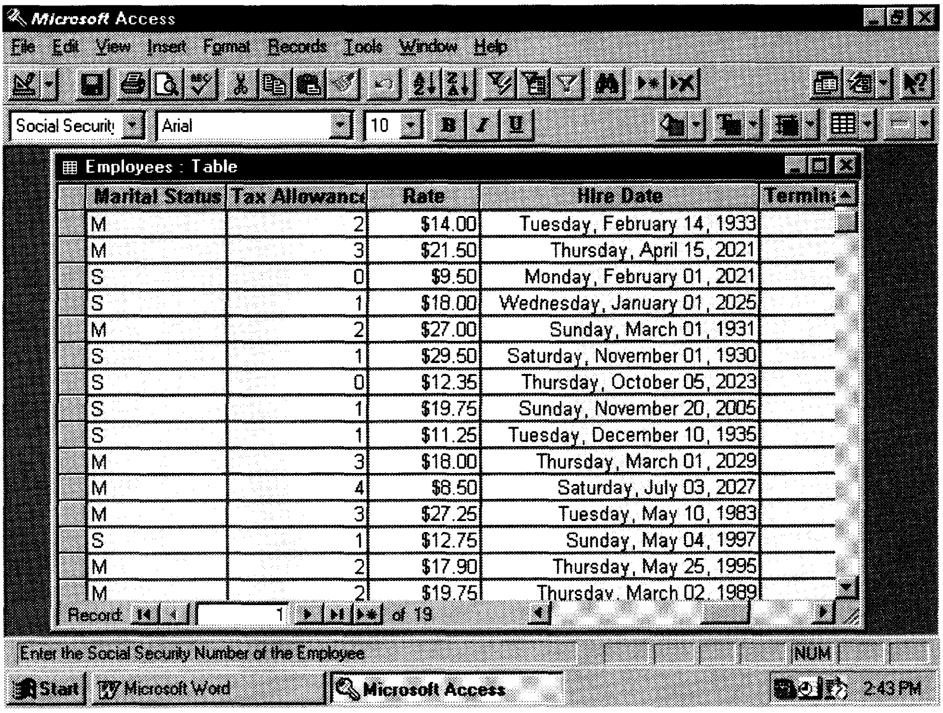
Databases

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.[11]

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."[12]

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

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



Marital Status	Tax Allowanc	Rate	Hire Date	Termin
M	2	$14.00	Tuesday, February 14, 1933	
M	3	$21.50	Thursday, April 15, 2021	
S	0	$9.50	Monday, February 01, 2021	
S	1	$18.00	Wednesday, January 01, 2025	
M	2	$27.00	Sunday, March 01, 1931	
S	1	$29.50	Saturday, November 01, 1930	
S	0	$12.35	Thursday, October 05, 2023	
S	1	$19.75	Sunday, November 20, 2005	
S	1	$11.25	Tuesday, December 10, 1935	
M	3	$18.00	Thursday, March 01, 2029	
M	4	$8.50	Saturday, July 03, 2027	
M	3	$27.25	Tuesday, May 10, 1983	
S	1	$12.75	Sunday, May 04, 1997	
M	2	$17.90	Thursday, May 25, 1995	
M	2	$19.75	Thursday, March 02, 1989	

Enter the Social Security Number of the Employee the century, when entering a two-digit year format. You can key in the four-digit year any time.[13]

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.

The definition of *y2k-compliant* can vary. 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 a significant one.

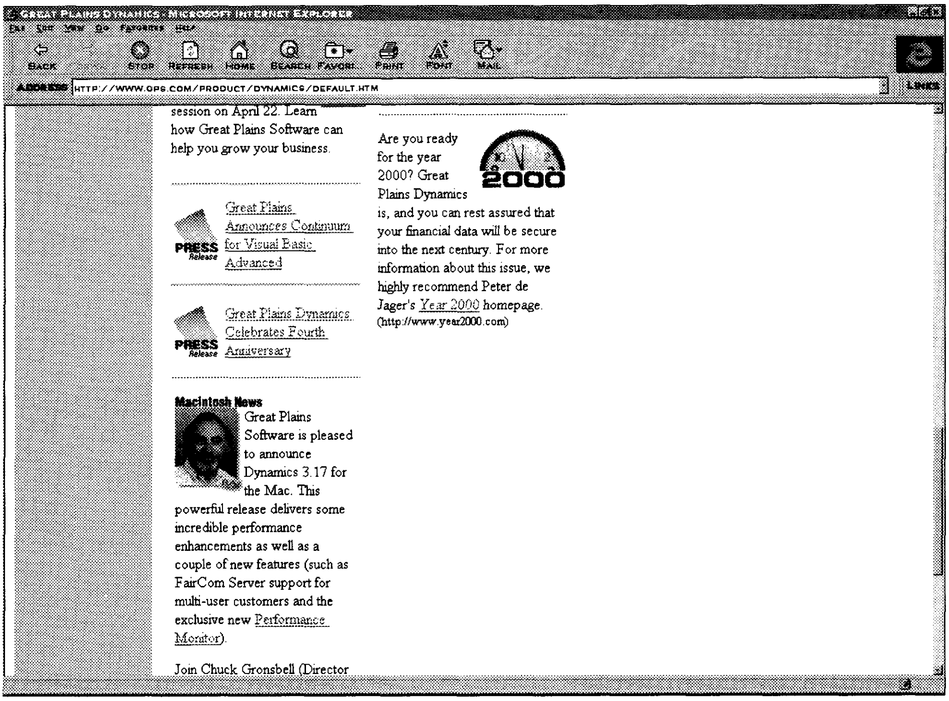
Accounting Systems

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 of 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 will ship September 26, 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's Dynamics is y2k-compliant. Its Web site has a white paper by Wayne Harding about the y2k issue.

FIGURE 3.9: GREAT PLAINS'S *DYNAMICS* IS Y2K-COMPLIANT



HTTP://WWW.GPS.COM/PRODUCT/DYNAMICS/DEFAULT.HTM

session on April 22. Learn how Great Plains Software can help you grow your business.

Great Plains Announces Continuum for Visual Basic Advanced

Great Plains Dynamics Celebrates Fourth Anniversary

Macintosh News

Great Plains Software is pleased to announce Dynamics 3.17 for the Mac. This powerful release delivers some incredible performance enhancements as well as a couple of new features (such as FairCom Server support for multi-user customers and the exclusive new Performance Monitor).

Join Chuck Gronsbell (Director

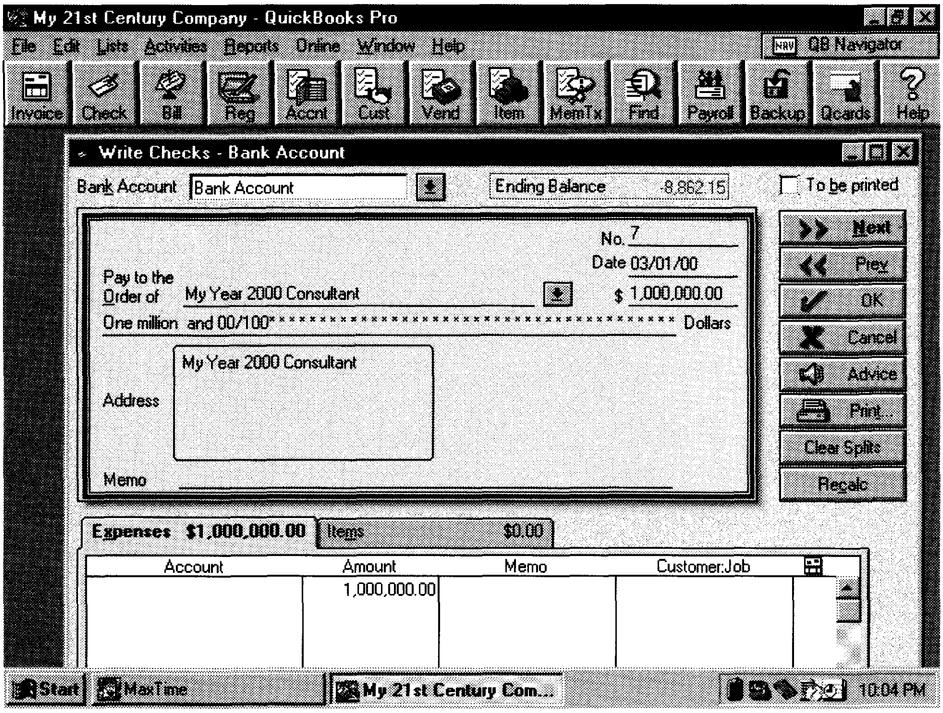
Are you ready for the year 2000? Great Plains Dynamics is, and you can rest assured that your financial data will be secure into the next century. For more information about this issue, we highly recommend Peter de Jager's Year 2000 homepage. (http://www.year2000.com)

- Macola offers its Version 7.0 Progression Series for its y2k solution.
- 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 are "strongly encouraged to subscribe now" 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. Their personal finance package Quicken is compliant with Version 6.0.
- SBT's Pro Series 3.2i lists "21st century support" in its product literature on the Web.
- Solomon III for Btrieve and Solomon IV for Windows are compliant according to the y2k page on Solomon's Web site.

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.

FIGURE 3.10: QUICKBOOKS 5 IS Y2K-COMPLIANT



Bank Account: Bank Account

Ending Balance -8,862.15

□ To be printed

No. 7

Date 03/01/00

Pay to the Order of My Year 2000 Consultant $ 1,000,000.00

One million and 00/100************************************* Dollars

Address: My Year 2000 Consultant

Memo

Expenses $1,000,000.00 Items $0.00

Account	Amount	Memo	Customer:Job
	1,000,000.00		

OTHER PACKAGED SOFTWARE

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

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

Network operating systems deserve a few sentences, too. *Windows NT Server* Versions 3.5 and 3.51 are y2k-compliant. Novell plans to complete its y2k project by the end of 1997. Its plan is to have y2k-ready products shipping after January 1, 1998.

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 the y2k in late 1996.

LEGACY SYSTEMS

The bulk of the effort of most y2k projects 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.

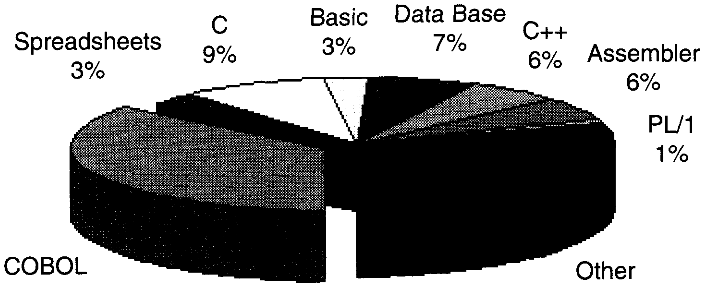
A y2k change in one program is a relatively simple maintenance task. The legacy system scenario quickly becomes complex as additional details are considered, such as the age of the programs and the lack of standards in those days. Consider the following points about the COBOL environment in the seventies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

- 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 seventies 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 seventies 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 half 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 less than 2 years 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 eighties. 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 on 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.[14] 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.
- 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 y2k.

FIGURE 3.11: PROGRAMMING LANGUAGES AND APPLICATIONS IN USE TODAY[15]



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 36 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 100 lines of code and makes one change, there will be a one in six chance of introducing another error in the code.[16]
- 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 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 eighties, 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 is primarily a mainframe problem, other platforms such as client/servers will need attention as well. A Morgan Stanley survey shows that there will be a large amount of work converting PC-based client/server systems.[17]

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

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

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 y2k, it could put out the wrong message on the wrong day. For a small business reliant on phone orders, that's a big deal. Billing details could be erroneous. Date-stamped messages in voice mail could also be incorrectly recorded.

If a noncompliant system is used in a city's 911 operation, any malfunction can become critical.[18]

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.[19] Both Nortel and Mitel have excellent y2k information on their Web pages. A quick search for "year 2000" 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 if 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.

Other business equipment that should be checked includes copiers and faxes. 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.[20] If you run a nursing home or hospital, these systems must be analyzed. The solution is to contact the vendor and find out if there is any corrective action to be taken on the system prior to y2k.

Elevators have gotten a lot of publicity. Supposedly, they will place themselves out of service at midnight y2k 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.

Medicine

Industries with products that can generate harm if they fail should be especially aware of the y2k implications. Medicine is one example. Hospitals, nursing homes, home health agencies, physician's offices, manufacturers of medical devices, and companies that sell or rent medical equipment should analyze their inventory. 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 any patients.[23] They can also contact the vendor and question them about compliance.

THE DEFENSE INDUSTRY

Defense is another industry with potential y2k risks. If the United States is involved in a conflict during the century change, its communications systems, transportation systems, and older equipment with embedded systems could show problems. Some of the software languages used to write defense systems, such as Jovial, are not widely used in business applications. That means it is harder to find individuals with the necessary programming skills that can make these changes.[24]

Manufacturers design some systems especially for the Department of Defense. These include the software embedded in weapons. These systems may no longer be actively produced. They may have been a part of a special military design and not available for general business use. The cost to fix this type of defense software can run as much as eight times the cost of normal business application software.[25]

Software glitches are not new to the Department of Defense. During a training exercise, screens went blank for mission planners trying to launch 2,700 air strikes. This glitch was caused by an expiration date in a software licensing manager program.[26]

THE BANKING INDUSTRY

The Federal Financial Institutions Examination Council (FFIEC) issued a strong statement to member banks in May 1997. It encouraged member banks to complete an inventory of systems and to have a plan in place by September 1997. Regulators will examine conversion efforts of banks through mid-1998. It further encouraged banks to be substantially complete with changes by the end of 1998.[27]

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

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 codes, 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 time-sensitive 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, a company must identify all of the electronic connections it has with external entities. A notification letter should be sent to each business partner asking when their 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 if 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.

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 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 1 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 y2k.

Even experts have been confused on the leap year issue. *Computerworld,* which has done a great job of covering the y2k issues for the information systems profession, ran a letter from a reader who cited the *Encyclopaedia Brittanica,* 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

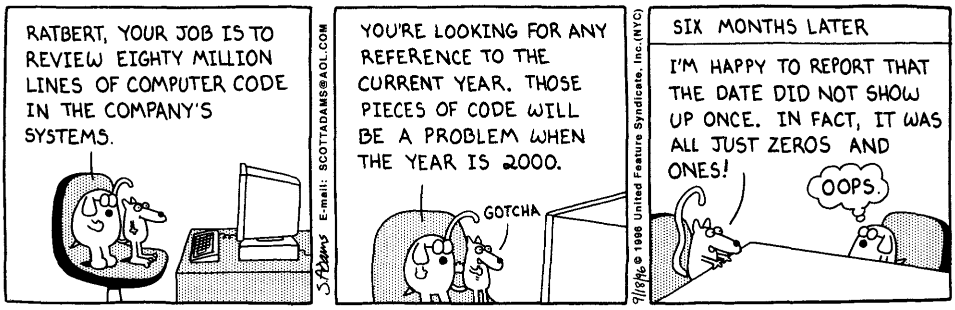
There is no silver bullet for the y2k. As you have already learned, the problem is too pervasive and too complex. Certainly there are tools that will reduce the length of a y2k project, maybe by as much as 30 percent. But there will be no silver bullet. Unless, of course, you are in the media. Here are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

1. From *The Dallas Morning News:* "Program Offers Quick Fix to Year 2000 Glitch." 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 15 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 this is no silver bullet.
2. From *The Wall Street Journal:* "Bob Bemer Aims Silver-Plated Bullet At Year 2000 Problem." 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 press and by companies that sell the products.

Section II

Year 2000 Issues


RATBERT, YOUR JOB IS TO REVIEW EIGHTY MILLION LINES OF COMPUTER CODE IN THE COMPANY'S SYSTEMS.
S.Adams E-mail: SCOTTADAMS@AOL.COM
YOU'RE LOOKING FOR ANY REFERENCE TO THE CURRENT YEAR. THOSE PIECES OF CODE WILL BE A PROBLEM WHEN THE YEAR IS 2000.
GOTCHA
9/18/96 © 1996 United Feature Syndicate, Inc.(NYC)
SIX MONTHS LATER
I'M HAPPY TO REPORT THAT THE DATE DID NOT SHOW UP ONCE. IN FACT, IT WAS ALL JUST ZEROS AND ONES!
OOPS.

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? A table from International Data Group identifies when chief executive officers (CEOs), chief financial officers (CFOs), and chief information officers (CIOs) first learned about the problem.[1]

%	*CEOs*	*CFOs*	*CIOs*
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.

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.

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

FIGURE 7.1: RISK OF Y2K PROBLEM TO ORGANIZATION

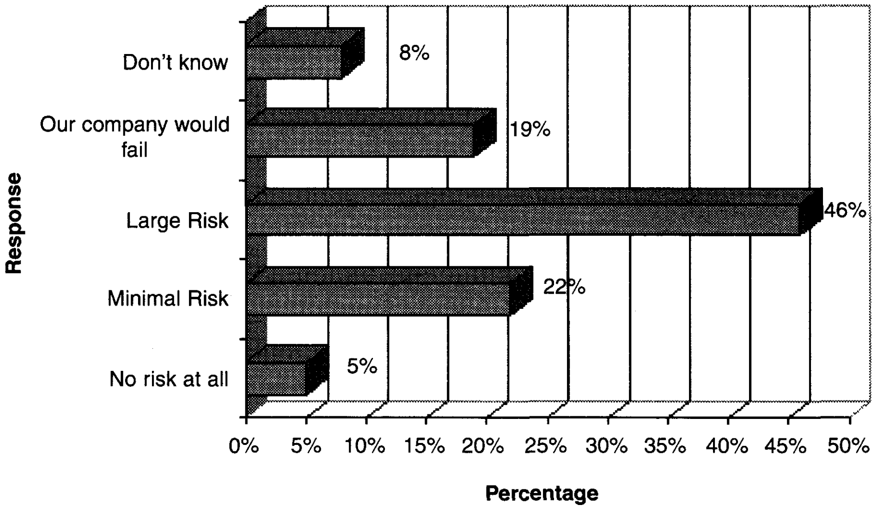
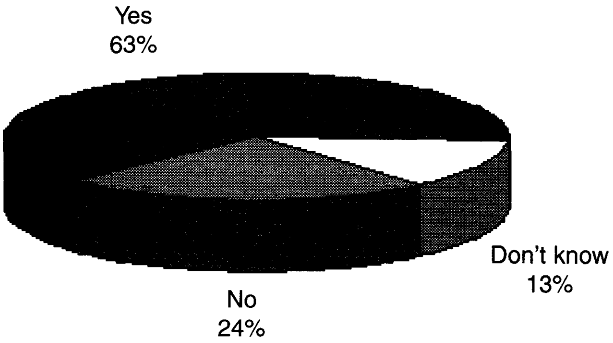


FIGURE 7.2: EXECUTIVE AWARENESS OF Y2K PROBLEM



1. How would you rate the y2k issue?

Response	***%***
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. Is your CEO and CFO aware of the issue?[3]

Response	***%***
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 performed an assessment to find out what the impact of the y2k would be on their systems.[4]

Based on these survey results, the Gartner Group statistic that half of all businesses won't be ready for the y2k and one-third of mission-critical systems won't be converted looks plausible.

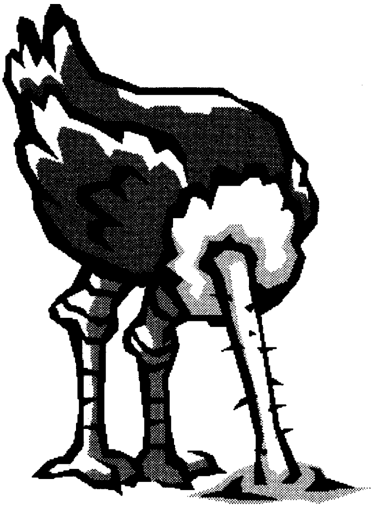
Michael Budzich of Gemko Information Group, Inc. in Buffalo, New York categorizes businesses into three groups:

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: the executives in denial, or as we say in East Texas, the ones that are frozen like deer in front of bright headlights.

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 on 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 are 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 impact 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 grief:

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 at 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.[5]
8. Ensure that the executive understands the pervasiveness and the magnitude of the problem.
9. Gain the executive's commitment as 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 except for the smallest of companies. All companies should be actively addressing the y2k dilemma.

Chapter 8: Accounting Issues

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 software to correct for the y2k problems should be treated as an expense in the current period.

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 over their estimated useful life. The American Institute of Certified Public Accountants's (AICPA's) Accounting Standards Executive Committee's proposed statement of position (SOP), *Accounting for the Costs of Computer Software Developed or Obtained for Internal Use,* provides new guidance in this area.

Software vendors will still follow FASB's Statement 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:

Impact on Profits (in $000,000)	***1998***	***1999***	***2000***	***2001***	***2002***
Cost to replace $10 million	($2)	($2)	($2)	($2)	($2)
Cost to fix $8 million	($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? There is currently nothing in the accounting literature that specifically addresses the reporting of future costs of modifying internal-use software. The Securities and Exchange Commission

(SEC) 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.

The Audit Issues Task Force (AITF) of the Auditing Standards Board (ASB) discussed the y2k dilemma and is considering whether formal interpretation of the auditing standards is necessary. It observed, during its continuing monitoring of the y2k dilemma, that the y2k issue did not pose any additional requirements for auditors. ASB chair Edmund R. Noonan said that a factor in the task force's discussion was the entity-specific nature of the problem. Different companies will face different magnitudes of the problem and will reach different solutions in addressing the issue. Although it is not a requirement, auditors might want to comment on y2k problems in establishing an understanding with the client on the nature and limitation of an audit of financial statements (including management's and the auditor's responsibilities) and in management letters or other similar communications with management and the board of directors.

CAVEAT

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

The 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 y2k.

The AICPA also plans to develop guidance for practitioners in a joint project with the Canadian Institute of Chartered Accountants (CICA).

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 y2k.

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?

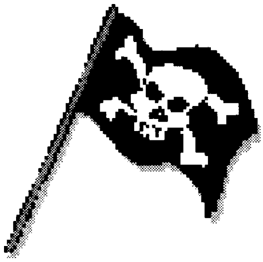
Capers Jones goes further. He estimates legal fees at $2 trillion and damages at $100 billion between 1997 and 2005. A Fortune 500 company, major bank, or insurance company should expect to shell out $750,000 per year for those 8 years.[2]

Six groups could suffer most and seek litigation:

- Clients whose finances have suffered
- Shareholders whose companies missed the deadline
- Estates of individuals who died, or individuals who were injured as a direct result of a y2k problem (See the sidebar "Software That Kills.")
- Customers with class-actions against software firms
- Customers of y2k consultants who missed the deadline
- Customers of hardware companies producing defective microcode[3]

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 compliance[4]



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 (IV) 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:

- An Iraqi 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-eighties 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 above 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 y2k 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 if 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 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. However, there may be some interesting litigation in this area from customers who are using 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 economically driven 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.

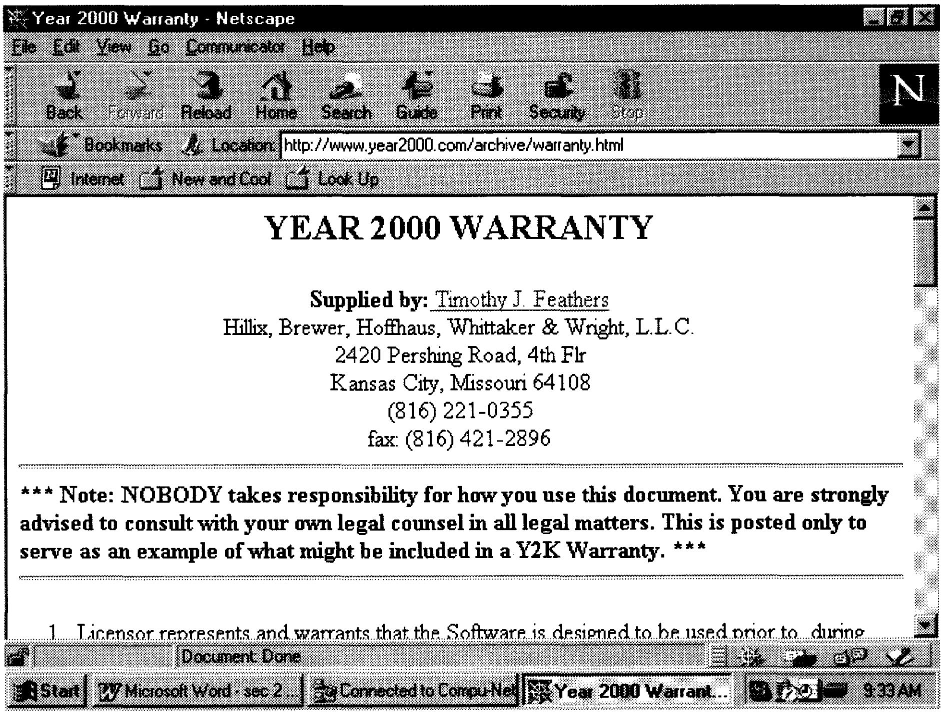
Today's buyers should insist on a warranty in all new contracts. The warranty shall state that the software will work before, during, and after the y2k and that it has y2k capabilities. Terms such as y2k compliance, leap year, and four-digit year format should be defined. The warranty should include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

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 amount 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.

Complications will arise. What if there is an unrelated bug in the software that is found? 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

FIGURE 9.1: Y2K WARRANTY


Year 2000 Warranty - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Guide Print Security Stop
N
Bookmarks Location: http://www.year2000.com/archive/warranty.html
Internet New and Cool Look Up
Document: Done
Start
Microsoft Word - sec 2...
Connected to Compu-Net
Year 2000 Warrant...
9:33 AM

YEAR 2000 WARRANTY

Supplied by: Timothy J. Feathers
Hillix, Brewer, Hoffhaus, Whittaker & Wright, L.L.C.
2420 Pershing Road, 4th Flr
Kansas City, Missouri 64108
(816) 221-0355
fax: (816) 421-2896

***** Note: NOBODY takes responsibility for how you use this document. You are strongly advised to consult with your own legal counsel in all legal matters. This is posted only to serve as an example of what might be included in a Y2K Warranty. *****

1. Licensor represents and warrants that the Software is designed to be used prior to, during

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 opening themselves to a harsher level of warranty should the system fail to live up to its certification.[12]

TO DATE

One y2k case has already been settled. A magazine publishing company sued a computer services firm for failing to supply updated software. It collected $4 million.[13]

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.[14]

Chapter 10: 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 a company's y2k plan is conducted by outside consultants to determine if 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, CA, that 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, VA, offers a certification on the *process* of y2k compliance. This is not a guarantee that everything will work in y2k. Thirty companies have signed up for the certifications, which cost $8,200.[4]

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 Resources

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 resources instead of part-time 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 20 analyst-years or more. This means that individuals have managed a 4-year project with a team of five people, or a 2-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 is 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.

The 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 y2k? 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 Resource?

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–$80,000, and Y2k consultants were earning $100,000–$150,000.[1] In early 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]
2. Consider offering benefits and perks, such as allowing individuals to work at home and on flex hours. Perks such as new PCs, casual dress, and technology training appeal to systems staff.

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. To find people late in the game, companies must become creative. They may resort to:

- Using offshore resources from India, Canada, or Ireland[3]
- 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).
- 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.
- 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.
- 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.[4]
- Disaster recovery managers should be involved to create contingency plans if portions of the project are not completed in time.[5]

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 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 resources 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:

- 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?

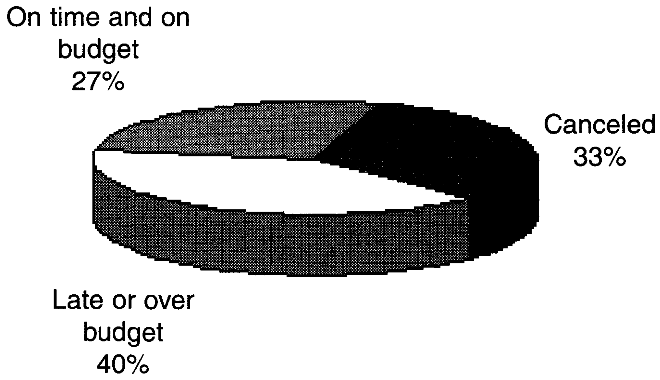
The Real Deadline

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:

1. The cost of everything related to y2k will be driven up greatly as January 1, 2000 approaches.
2. Experienced personnel may be committed on other projects and unavailable.
3. Products or services to correct the y2k problem may not be able to be manufactured or delivered in time for those who wait.
4. There will not be enough time to complete all of the steps necessary to correct the problem if you start too late.
5. 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.

FIGURE 12.1: PROJECT STATUS



The y2k compliance work needs to be done much sooner than that. What if account expiration date cycles are 3 years? What if forecasting systems project 5 years into the future? What if loan payments are spread over 20 years? Chances are, you've already had y2k problems with systems that track this type of information.

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.

THE OWNERSHIP ISSUE

Deep in the bowels of a company's computers, there may be a system that is hiding, 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 of who owns these systems are not easily answered unless they've been worked out before. The ownership questions boil down to two issues: who is going to pay for the changes, and who has responsibility for authorizing changes to the system? 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 decided to blow 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 we 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 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.

BUDGETING FOR THE Y2K

Most companies are taking staff and dollars away from existing projects to fund y2k efforts. In a Cap Gemini poll, companies said they are spending 20 percent or less of their 1997 IT budgets on the y2k project. That percentage is expected to rise to between 20 and 40 percent over the next three years, however.[2]

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.[3]

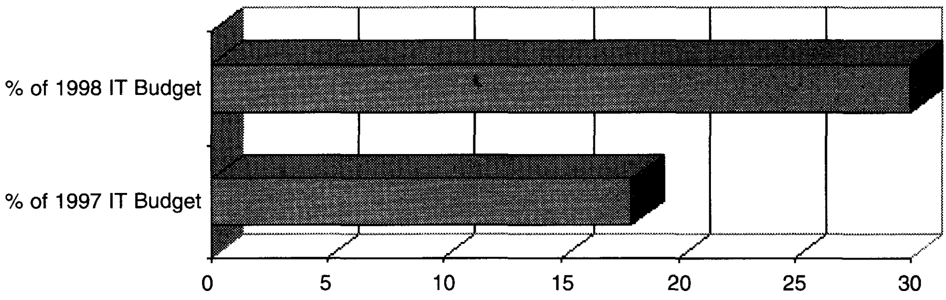
How much will it cost to fix your y2k problems? Gartner Group provides averages from $1.10–$1.65 per line of code, depending on what modification option you use. Windowing costs $1.10; date expansion costs $1.65. (See chapter 18 for an explanation of the various modification options.) The Gartner Group cautions that these figures should only be used as first guesses, and they warn of an error rate of 40 percent.[4] A Morgan Stanley paper lists line-of-code estimated costs from $0.90–$1.70.[5]

"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, a line of code is of varying lengths. Capers Jones argues that function points are a much better measurement than lines of code.[6]

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

- Inputs
- Outputs
- Inquiries
- Logical files
- Interfaces[7]

FIGURE 12.2: Y2K BUDGET AS A PERCENT OF IT BUDGET



A typical corporate system portfolio contains 500,000 function points. The Microsoft Word program has 5,000 function points. It takes 2 calendar years to develop an entire system the size of 1,000 function points.[8]

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–$40 per function point. The cost per function point rises as the project size increases.[9]

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 detail 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

Three 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, the y2k doesn't have to be all negative. I'll complete this chapter with a positive spin on y2k.

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 bankruptcy for some percentage of firms is inevitable.

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. Four examples are listed below.

1. They may not be as aware of the y2k dilemma.
2. Small companies do not have backups to their systems like large companies do.
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.[3]

Nevertheless, companies with less than 100 employees own less software than other companies and will generally have less of a problem in magnitude than 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 5–7 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 post-y2k.[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. The Gartner Group estimates the currency conversion will cost $100 billion worldwide.[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.

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.[9]

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. As noted, some businesses will go bankrupt from not preparing for the y2k. This should present some acquisition opportunities for the survivors in the coming year or two.

Section III

Year 2000 Solutions

HERE'S MY INVOICE FOR FIXING YOUR "YEAR 2000" COMPUTER PROBLEMS.
www.unitedmedia.com
S.Adams
AAAEEE!!!
ooo
9/9/96 © 1996 United Feature Syndicate, Inc.(NYC)
...SO HIS HEAD SPUN, BUT IT DIDN'T EXPLODE?
YEAH. I GUESS I LEFT SOME MONEY ON THE TABLE.

Dilbert reprinted by permission of United Features Syndicate, Inc.

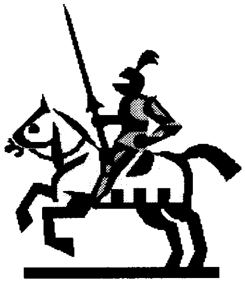
This section will present a step-by-step approach to conquering the y2k dilemma in your firm. The steps we will follow include:

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

Each of these steps will be discussed in detail in the following chapters. Three more chapters will provide more detailed information about 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 steering committees 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 quarterly, then monthly 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 CEO or chief operating officer (COO).

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.

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

Two tasks in the project initiation phase deserve extra attention.

1. The company's purchasing policy should be modified immediately to acquire only y2k-compliant goods and services.
2. A program to educate all employees on the y2k dilemma should be completed.

These tasks are discussed below.

UPDATE YOUR PURCHASING POLICY

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 for 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 below:

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 y2k 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 the above policy to their own and clients' current purchasing policies. This step is one that should be implemented immediately.

EDUCATE EVERYONE IN THE COMPANY

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. The contacts so far surely made the company sound like it didn't have a clue about y2k. About a week later, my customer service representative gave me the name of a consultant who 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, either a newsletter or intranet posting 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

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 for the detailed project plan and budget (unless there is unlimited time and money to finish)

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. Information about each system should be gathered:

- Name and description of the system.
- Do you plan to have the system in y2k? (If no, list 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?
- What is the origin of the system?
  - — If the system was purchased from a vendor, list:
    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 components of the system.
- Is the system documented or is there anyone in the company that knows the system?
  - — List particulars.

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

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 for an abbreviated checklist.) One project manager commented in an Internet discussion group, "I am looking at everything that is plugged in." 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:

- 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 management[1]

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 if 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 three groups of systems:

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

Systems that you do not plan to have in y2k 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 in those years, 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, a few 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 testing is not necessary and no further action is required.

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. 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 its vault containing the safe deposit boxes to determine if an upgrade is necessary.
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.

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 are:

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: available for all systems.
6. 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.

Replace the System

The replacement option works best if:

- 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 an existing system include:

- Additional hardware may need to be purchased to run the new system
- Replacement costs may end up being more than 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 2 or more years.

Vendors that offer y2k-compliant systems are finding that their sales are rising. SAP, PeopleSoft, and SAS are three examples of vendors 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 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.[2]

Other vendors are offering the y2k upgrade free provided you are a member of their maintenance program. These vendors are selling a lot of maintenance programs as 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 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 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 like 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 if 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 if 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 many cases. Many businesses will not have the luxury of making every system in their organization 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

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:

- Macros 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 the 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.

SET PRIORITIES

Now that you have an understanding of what alternatives can be chosen, it is useful to assign a tentative alternative to each system. You may want to assign two alternatives 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 the 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.

This is a form of triage for the types of systems that your company depends on. 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 6 months to kick around alternatives any more. It is important to fast track your decisions and get started as fast 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. So many companies have busted their schedules already on this step by overanalyzing alternatives and waiting until monthly meetings roll around to push ideas around some more. No one has this luxury any longer.

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 out 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 our friends. Included in this section is chapter 19 on consultants and what services they are offering. 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 feel they have adequate expertise in house to solve the problem
- Many companies think the problem is too critical and should not be delegated

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 discusses the tools that are available for y2k projects.

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, then a brief comment will be made 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 routine 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 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; and 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.

UPGRADING THE SYSTEM

For systems that can be upgraded, a new software package version must be obtained. 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 a 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 60 percent of the time of a y2k project. The time it will actually take will depend on how much legacy code is being changed as well as how many new systems are being installed versus upgraded. A company that can upgrade its packaged systems and become y2k-compliant will not need to spend as much time testing as a company that is changing its legacy code.

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 of test scenarios, or test scripts, that should be created as part of a test plan. 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 2001–2010 year range for correct processing.
- Check to see that month and day or month and year are never confused.
- Check leap year days.
- Check a set of data in 2000 for correct processing.
- Check date calculations of birthdays in both centuries.
- Check the processing of transactions across the 1999–2000 time frame.
- Check date calculations that span 2 years, such as an order date in 1999 and a shipping date in 2000.
- Check multiyear transactions, such as loans that span 30 years, for correct processing.

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.
- *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, as 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 for more information on vendor services.

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.

The degree of testing quality can vary. If the software can impact 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. 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.

The testing portion of the project is very challenging and very hectic. Usually, the project is behind schedule and the time allotted testing is compressed to try to bring 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. Program changes and retesting (or in the case of packaged software, vendor notification) must be made quickly to correct problems. The testing process is iterative. Although testing is hectic and fast-paced, 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 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 thing is true with system documentation for the users. Minor changes may be needed with systems that were made compliant. Replacement systems and system upgrades will require new manuals. The documentation should be ready for distribution prior to installation, or prior to 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 data 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.

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 that is now in charge of maintaining the system.

One last project task, which is hardly ever done because the project is late or the team is burned out, is to perform 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.

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 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 if 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 impacts 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 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 a lot of dates or 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 for which the program will return the correct answer.

Windowing code looks like this (in English):

```
IF THE DATE IS LESS THAN "80"
  MAKE "20" BE THE CENTURY
OTHERWISE (IT WILL BE GREATER THAN OR EQUAL TO "80" SO)
  MAKE "19" BE THE CENTURY
```

This code will return correct answers for years between 1980 and 2079. 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 but when programs work with multiple dates at once 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 when 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 a great deal of 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.

DATE COMPRESSION

Date compression 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 data versus its 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 involves translating the date by aging it 28 years. Why 28 years? Because 28 years ago, the calendar was identical to the current year. Thus, August 5 was a Tuesday in both 1969 and 1997. 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 28 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 28 years, the user of this method gains roughly 30 more years to make a decision about what they are going to do in the "y2k."

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.[1]

Chapter 19: Y2k Consultants

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

Ernst & Young (E&Y) has a nice Web presence on its y2k services. It provides a turnkey solution, beginning with program management. E&Y will help a company identify key players, partners, risks, and scope, and will coordinate the effort of structuring the management team for the effort. It provides a methodology for proceeding through the inventory and analysis phases and understands the hugeness of the project management effort. The project steps that are followed include:

- Program management
- Project management
- Accelerated scoping
- Assessment and planning
- Business Partners Y2k Readiness Management
- Accelerated design and technical solution
- Acceptance testing
- Implementation
- Infrastructure and support

The Business Partners Y2k Readiness Management step involves creating a plan to measure the level of compliance of the entities that the company does business with. This separate step emphasizes its importance. E&Y analysts attempt not only to monitor the progress of the external entities but also to speed their compliance.

E&Y offers both programmers and machine cycles at its *conversion factory* for the coding and testing phases of the project. It has developed alliances with an overseas programming facility, Tata Consultancy Services, and a tools vendor, Crystal Systems Solutions, a subsidary of Formula Systems, Ltd. Tata Consultancy Services is a well-known software development company that employs over 8,500 programmers in India. Crystal Systems Solutions developed a conversion tool called C-MILL.

Price Waterhouse has partnered with Computer Horizons to provide y2k solutions to clients. Computer Horizons uses an automated tool called Signature 2000 to provide a full life cycle solution. The Price Waterhouse Technology Center and Advanced Software Engineering Centers provide resources to clients on y2k and other technology services.

Coopers & Lybrand (C&L) calls its methodology Diagnostic 2000. It utilizes a scanner that scans a company's program libraries and looks for dates. The results are fed to an econometric model which produces reports on time and cost estimates. C&L's Computer Assurance Services (CAS) group and the consulting group, Solutions Through Technology (STT) work together to offer Transition2000 project management services. They work with businesses of all sizes and list financial services, insurance, and manufacturing companies as current clients. A spokesperson for the firm sees the projects evolving as time gets closer to the due date. Now the projects are focused on the diagnostic, inventory, and analysis phases. In 1998–1999, the emphasis should swing around to coding and testing services.

A Peat Marwick brochure describes a Year 2000 Conversion Factory in Montvale, New Jersey, which offers consultants, tools, and machine time. In addition to Global Year 2000 services that include assessment and testing, the firm offers Independent Verification and Validation (IV&V). This service provides an independent analysis of a y2k project whether led internally or by third-party vendors. IV&V is intended to provide the client with information about the status of the project and whether the project is estimated to be complete in time. In its brochure, KPMG lists that it was founded in 1897 and that it helped their clients transition to the year 1900.

Deloitte and Touche is partnering with Data Dimensions to offer its y2k services. It is using a tool from Data Dimensions called ARDES 2K that provides analysis and a project methodology for y2k projects. Data Dimensions is a technology consulting company and has been offering y2k services since 1991.[1]

Andersen Consulting has forty to fifty active y2k engagements with a staff of 700–1,000 consultants. It has centers in the Philippines and in Madrid, Spain. The firm emphasizes meeting existing clients' needs and will limit the growth of the y2k services it offers. It is turning away work and is selective in the work it accepts, although it does see a revenue opportunity as well as improved client relationships with y2k consulting.[2]

Very few mid-sized accounting firms are offering y2k work, according to a random sample of telephone interviews conducted in August 1997. Grant Thorton does offer technology services to its clients, including client/server technologies. But it is not doing any y2k work. John Montgomery, National Director for Management Consulting of Grant Thorton, said since most of the work was in legacy systems, that wasn't its direction. He said that Grant Thorton "didn't want to build up a staff in legacy systems, because we don't think that's where the market is going."

One director of information systems at a mid-sized accounting firm in the Midwest thought that they might start offering the service in late 1997 or early 1998. Most of their clients were small businesses, where a y2k project would not have the same impact as on larger businesses.

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 issued by the partnership this year 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.

Very few small firms offer y2k consulting. A search for "CPA" and "year 2000" on the Internet turned up nothing. It is not too early 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 3-year technology plan or an assignment to evaluate new software. As of summer 1997, I have completed several y2k plans for small companies.

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 for its y2k work.[3]

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.[4]

IBM has the largest staff of y2k consultants, 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 compliance[5]

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.[6]

Both IBM and Andersen 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 efforts basis.[7]

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?
2. Have you or has 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?

A majority of yes answers probably means that 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.

1. Decide which part of the project needs 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. Vendor has been in business for 10 years or more.
   b. Vendor has completed similar projects successfully. (Ask for references.)
   c. The project manager has more than 15 years of experience in the industry and has been with the vendor for more than 5 years.
   d. Each member of the programming team has at least 5 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. Vendor can get along with key management, customers, and 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.[8]

4. Screen the list of vendors with a phone interview checklist or other method. Narrow the list down to five or less.

5. Depending on the size of the project, you can choose to send a request for proposal (RFP) now or in lieu of the above 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 above.

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:
   - *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, 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.* Cost[9]

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 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 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 above list can be narrowed to suit a smaller-sized 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

A few hundred 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.

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. These tools are not 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.

Date-identification tools sniff out 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 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.[1]

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.[2]
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.
4. Mainware, Inc.'s HourGlass 2000. Kemper Insurance's data administration manager, Mitch George, is happy with the results of this tool.[3]
5. MicroFocus, Inc.'s Revolve/2000. Kemper uses this tool for code analysis.[4]
6. Global Software, Inc.'s Giles.[5]
7. AstraTek, Inc. sells a product called *VisualAudit*. It analyzes dates on Excel worksheets, Access, and Visual Basic programs. Stephanie Moore, an analyst at Giga Information

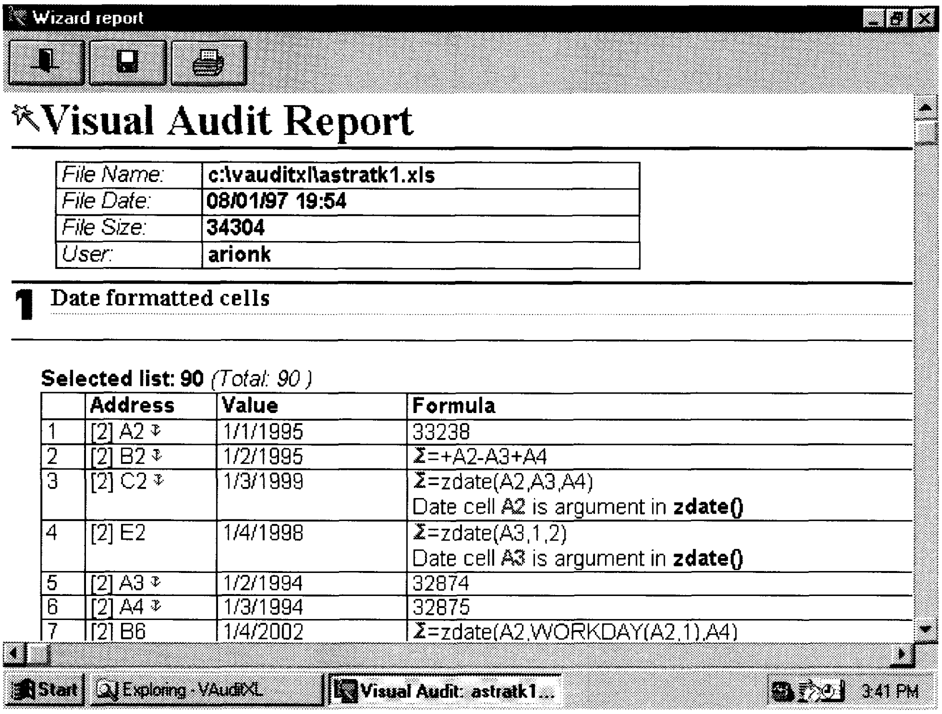
Group, said the tool has been purchased by Wall Street types that maintain powerful spreadsheets.[6] Check out AstraTek at http://www.astratek.com.

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.[7]

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.[8] Formal Systems, Inc. says two statistics on accuracy are important to know: false alarms (or false negatives), when code is changed

FIGURE 20.1: A REPORT FROM VISUAL AUDIT SOFTWARE



Visual Audit Report

File Name:	**c:\vauditxl\astratk1.xls**
File Date:	**08/01/97 19:54**
File Size:	**34304**
User:	**arionk**

1 Date formatted cells

Selected list: 90 *(Total: 90)*

	Address	Value	Formula
1	[2] A2 ⇕	1/1/1995	33238
2	[2] B2 ⇕	1/2/1995	Σ=+A2-A3+A4
3	[2] C2 ⇕	1/3/1999	Σ=zdate(A2,A3,A4) Date cell A2 is argument in **zdate()**
4	[2] E2	1/4/1998	Σ=zdate(A3,1,2) Date cell A3 is argument in **zdate()**
5	[2] A3 ⇕	1/2/1994	32874
6	[2] A4 ⇕	1/3/1994	32875
7	[2] B6	1/4/2002	Σ=zdate(A2,WORKDAY(A2,1),A4)

that didn't need to be, and misses (false positives), when 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.[9]

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.[10]

You should find out if 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.[11]
3. Peritus Software Services, Inc.'s AutoEnhancer/2000.
4. Computer Associates' CA-Realia II Workbench, CA-Visual Realia, CA-Verify, CA-Librarian, 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 Tools.[12]

7. Forecross Corporation converts C++ code.
8. Software Emancipation Technology, Inc.'s Discover Tool converts C and C++ applications.[13]

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 comparison utilities, date utilities, test data generators, and problem identification tools.

Vendors with tools in this category include:

1. Compuware's Xpediter+, 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 Tools.[14]

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.[15]
3. Intersolv, Inc.'s PCVS Version Manager.[16]

As the y2k gets closer, more tools will appear. Capers Jones estimates that there are 300 tools already available, and 20–25 new vendors are entering this market each month.[17] 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 a maintenance vendor it is, says Bruce Hall, research director at Gartner Group.[18] Not all y2k tools will have a track record. When evaluating tools, consider the following:

- Make sure the vendor has a tool that supports the programming languages in your business.
- With search tools, find out what percentage of y2k instances the vendor claims it will find.
- With repair tools, find out what percentage of y2k instances the vendor claims will be fixed.
- 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?[19]
- 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–$750,000, which includes tools and training, or 5–10 percent of their y2k project costs.[20] (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. Three online discussion groups are worth mentioning as a source of tips from the front line: individuals who have the daring job of assuring that the rest of us experience the century change unscathed. They are:

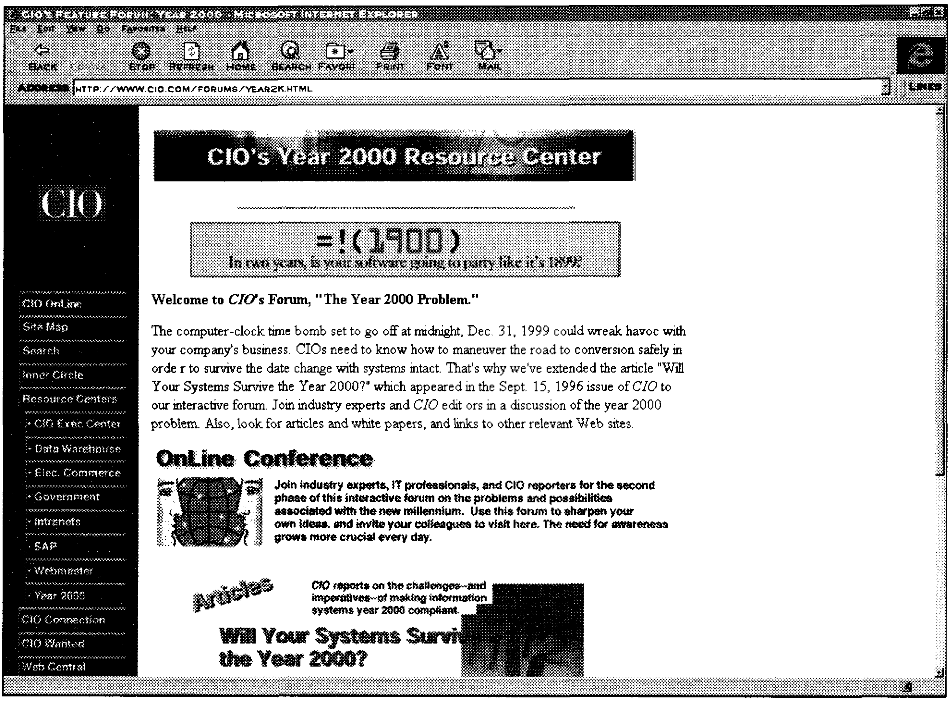
- *CIO Magazine*'s Year 2000 Online Forum. This discussion group features industry leaders on the topic and is contained within *CIO Magazine*'s Web Site's Year 2000 Resource Center. Find it at http://www.cio.com.
- Best Practices Web site sponsored by the Society for Information Management (SIM) Year 2000 Working Group. Find it at http://www.simnet.org.
- The Usenet news group, comp.software.year-2000.

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.

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, inventory checklists, and receive a list of CPA firms that provide y2k services. (If you would like to be listed, contact sandi@cyberrramp.net.)

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, his Web site should be the one. You can find it at http://www.year2000.com.

FIGURE 21.1: CIO MAGAZINE'S YEAR 2000 RESOURCE CENTER



HTTP://WWW.CIO.COM/FORUMS/YEAR2K.HTML

CIO's Year 2000 Resource Center

=!(1900)
In two years, is your software going to party like it's 1899?

Welcome to *CIO*'s Forum, "The Year 2000 Problem."

The computer-clock time bomb set to go off at midnight, Dec. 31, 1999 could wreak havoc with your company's business. CIOs need to know how to maneuver the road to conversion safely in orde r to survive the date change with systems intact. That's why we've extended the article "Will Your Systems Survive the Year 2000?" which appeared in the Sept. 15, 1996 issue of *CIO* to our interactive forum. Join industry experts and *CIO* edit ors in a discussion of the year 2000 problem. Also, look for articles and white papers, and links to other relevant Web sites.

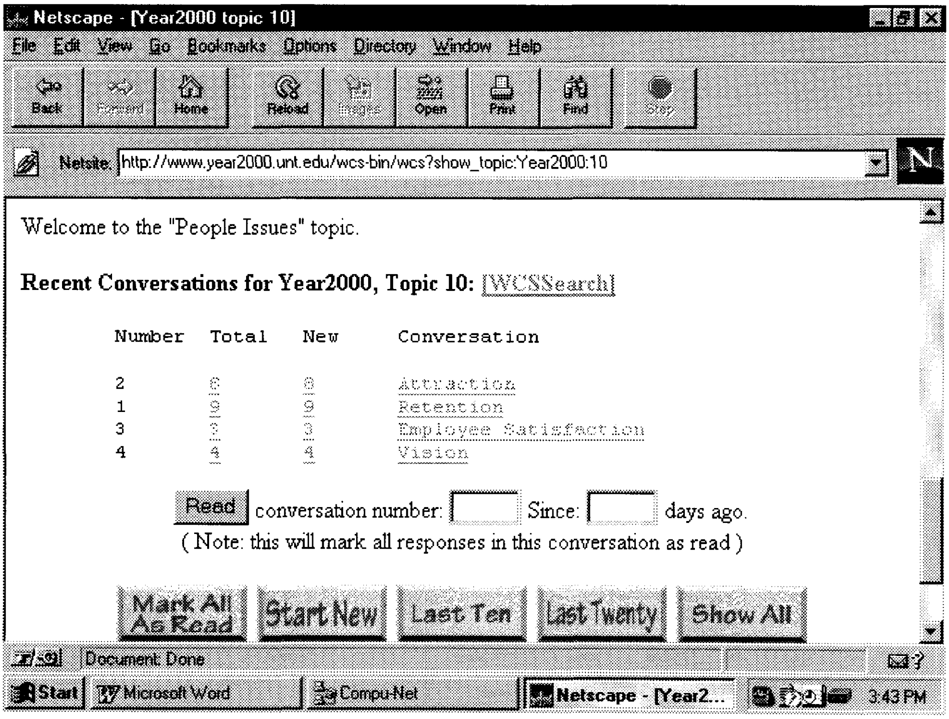
OnLine Conference

Join industry experts, IT professionals, and CIO reporters for the second phase of this interactive forum on the problems and possibilities associated with the new millennium. Use this forum to sharpen your own ideas, and invite your colleagues to visit here. The need for awareness grows more crucial every day.

CIO reports on the challenges--and imperatives--of making information systems year 2000 compliant.

Will Your Systems Survi the Year 2000?

FIGURE 21.2: A THREAD ON SIM'S YEAR 2000 WORKING GROUP'S BEST PRACTICES WEB SITE



http://www.year2000.unt.edu/wcs-bin/wcs?show_topic:Year2000:10

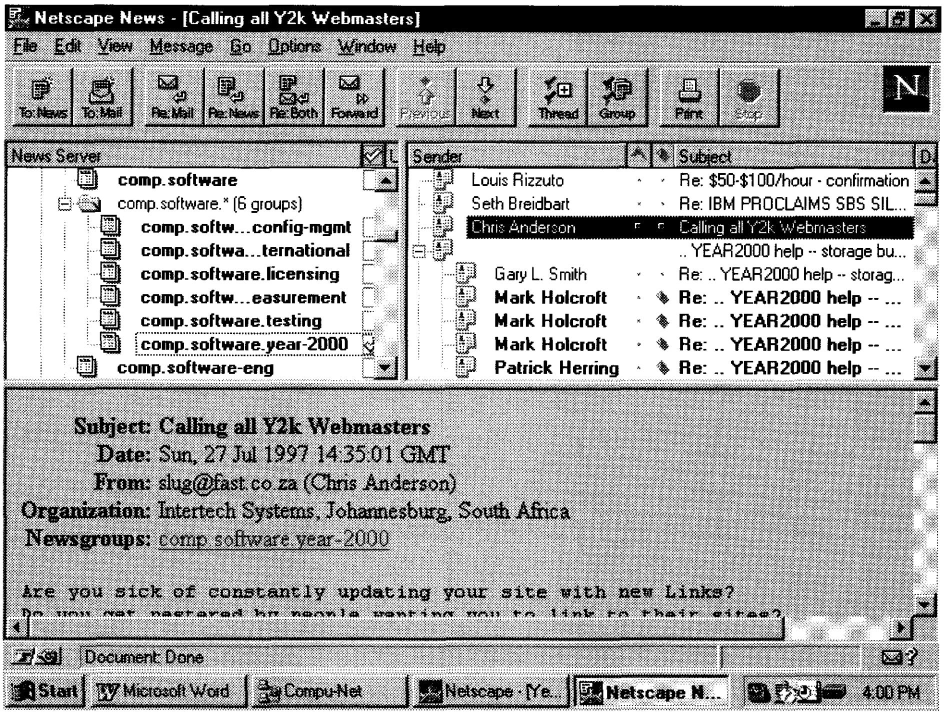
Welcome to the "People Issues" topic.

Recent Conversations for Year2000, Topic 10: [WCSSearch]

Number	Total	New	Conversation
2	8	8	Attraction
1	9	9	Retention
3	3	3	Employee Satisfaction
4	4	4	Vision

Read conversation number: ▭ Since: ▭ days ago.
(Note: this will mark all responses in this conversation as read)

FIGURE 21.3: A THREAD ON USENET'S COMP.SOFTWARE.YEAR-2000 NEWS GROUP



Sender	Subject
Louis Rizzuto	Re: $50-$100/hour - confirmation
Seth Breidbart	Re: IBM PROCLAIMS SBS SIL...
Chris Anderson	Calling all Y2k Webmasters
	.. YEAR2000 help -- storage bu...
Gary L. Smith	Re: .. YEAR2000 help -- storag...
Mark Holcroft	Re: .. YEAR2000 help -- ...
Mark Holcroft	Re: .. YEAR2000 help -- ...
Mark Holcroft	Re: .. YEAR2000 help -- ...
Patrick Herring	Re: .. YEAR2000 help -- ...

Subject: **Calling all Y2k Webmasters**
Date: Sun, 27 Jul 1997 14:35:01 GMT
From: slug@fast.co.za (Chris Anderson)
Organization: Intertech Systems, Johannesburg, South Africa
Newsgroups: comp.software.year-2000

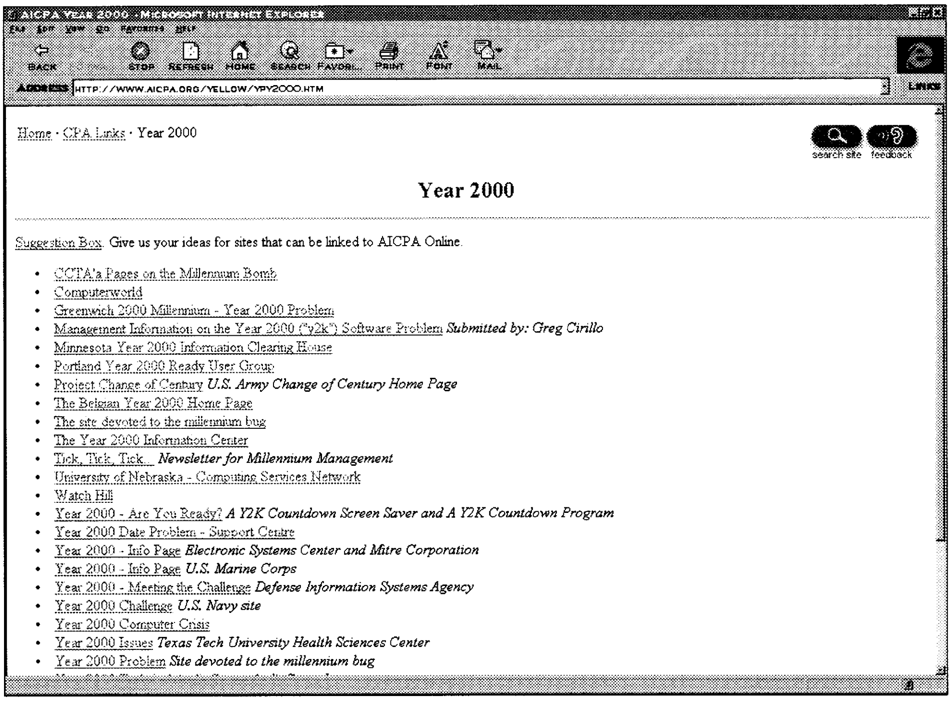
```
Are you sick of constantly updating your site with new Links?
Do you get pestered by people wanting you to link to their sites?
```

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



HTTP://WWW.AICPA.ORG/YELLOW/YPY2000.HTM

Home · CPA Links · Year 2000

Year 2000

Suggestion Box. Give us your ideas for sites that can be linked to AICPA Online.

- CCTA'a Pages on the Millennium Bomb
- Computerworld
- Greenwich 2000 Millennium - Year 2000 Problem
- Management Information on the Year 2000 ("y2k") Software Problem *Submitted by: Greg Cirillo*
- Minnesota Year 2000 Information Clearing House
- Portland Year 2000 Ready User Group
- Project Change of Century *U.S. Army Change of Century Home Page*
- The Belgian Year 2000 Home Page
- The site devoted to the millennium bug
- The Year 2000 Information Center
- Tick, Tick, Tick... *Newsletter for Millennium Management*
- University of Nebraska - Computing Services Network
- Watch Hill
- Year 2000 - Are You Ready? *A Y2K Countdown Screen Saver and A Y2K Countdown Program*
- Year 2000 Date Problem - Support Centre
- Year 2000 - Info Page *Electronic Systems Center and Mitre Corporation*
- Year 2000 - Info Page *U.S. Marine Corps*
- Year 2000 - Meeting the Challenge *Defense Information Systems Agency*
- Year 2000 Challenge *U.S. Navy site*
- Year 2000 Computer Crisis
- Year 2000 Issues *Texas Tech University Health Sciences Center*
- Year 2000 Problem *Site devoted to the millennium bug*

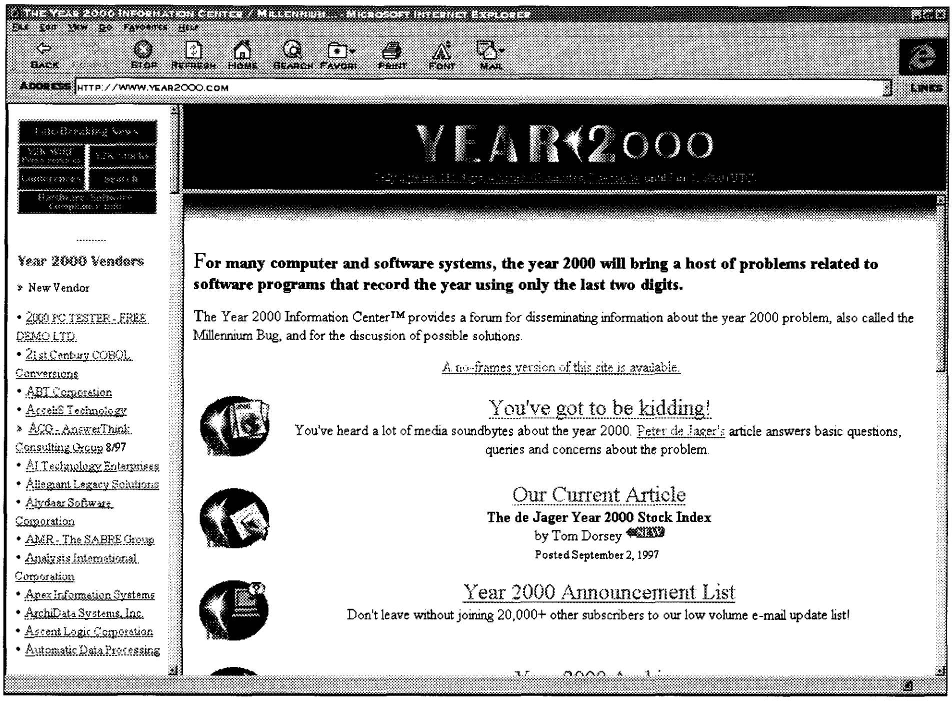
PROTECTING YOURSELF FROM THE FAILURES OF OTHERS

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. 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 if the project will fail 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, you can research competitors before the date approaches and decide on a contingency plan. If there are no competitors, you can try to time purchases and actions so that the y2k problem will have minimum personal

FIGURE 21.5: THE YEAR 2000 INFORMATION CENTER


THE YEAR 2000 INFORMATION CENTER / MILLENNIUM... - MICROSOFT INTERNET EXPLORER
FILE EDIT VIEW GO FAVORITES HELP
BACK
STOP
REFRESH
HOME
SEARCH
FAVORI
PRINT
FONT
MAIL
ADDRESS
HTTP://WWW.YEAR2000.COM
LINKS
YEAR2000

Year 2000 Vendors

- New Vendor
- 2000 PC TESTER - FREE DEMO LTD.
- 21st Century COBOL Conversions
- ABT Corporation
- Accel8 Technology
- ACG - AnswerThink Consulting Group 8/97
- AI Technology Enterprises
- Allegiant Legacy Solutions
- Alydaar Software Corporation
- AMR - The SABRE Group
- Analysts International Corporation
- Apex Information Systems
- ArchiData Systems, Inc.
- Ascent Logic Corporation
- Automatic Data Processing

For many computer and software systems, the year 2000 will bring a host of problems related to software programs that record the year using only the last two digits.

The Year 2000 Information Center™ provides a forum for disseminating information about the year 2000 problem, also called the Millennium Bug, and for the discussion of possible solutions.

A no-frames version of this site is available.

You've got to be kidding!

You've heard a lot of media soundbytes about the year 2000. Peter de Jager's article answers basic questions, queries and concerns about the problem.

Our Current Article

The de Jager Year 2000 Stock Index

by Tom Dorsey NEW

Posted September 2, 1997

Year 2000 Announcement List

Don't leave without joining 20,000+ other subscribers to our low volume e-mail update list!

impact. There will not have to 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 Y2k

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 take. When a basic assumption is not followed, such as taking into account the y2k, 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 not be a need to examine calculations, for example), but it would be a big one. Every phone number field would have to be found and expanded. When 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 20 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, you have less than 800 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:

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

External Interface Systems

- □ EDI (electronic data interchange) 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

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

OFFICE SYSTEMS

- □ Credit card systems
- □ Copiers
- □ Fax machines
- □ Telephone systems
- □ Mobile telephones
- □ VCRs and other video projection equipment
- □ Time clock systems

BUILDING SYSTEMS

- □ Heating and cooling systems
- □ Lighting systems
- □ Fire control systems
- □ Security monitoring systems
- □ Safes and vaults
- □ Card entry systems
- □ Elevators
- □ Parking systems (gate entry, ticketing and meters)
- □ Other building maintenance systems

SPECIALIZED INDUSTRIES

- □ Health equipment for hospitals and physicians' offices
- □ Automated teller machines for banks
- □ Process control systems for manufacturing plants
- □ Traffic systems for transportation industries

Appendix B: Sample Project Plan

Project Initiation

1. Identify executive sponsor
2. Identify project manager
3. Change acquisition/purchasing policy to assess future purchase of y2k-compliant goods and services only
4. Determine project methodology and project management tools
5. Create initial project plan and begin approval process
6. Create project standards
7. Determine whether a project office is appropriate and if needed, create one
8. Hire staff
9. Begin internal awareness campaign. This may include sending a newsletter or conducting training sessions with all employees in the organization
10. Complete project schedule and budget for assessment stage

Assessment

1. Determine if the organization will benefit from automated tools in this step. If so, analyze and purchase them
2. Select and acquire inventory tracking software
3. Begin an inventory of systems
4. Send vendor notification letters for packaged systems
5. Send letters to customers and suppliers with systems that interface with your systems
6. Complete the inventory
7. Begin an analysis of options with each major system listed in the inventory
8. Determine whether each system will be fixed, replaced, updated, or scrapped

9. 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
10. 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
11. 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
12. Prioritize the systems and schedule the various initiatives
13. Complete project schedule and budget for renovation stage

RENOVATION

1. For systems that will be eliminated, pull the plug
2. For systems that will be fixed, select and acquire tools that will be used during conversion
3. Hire staff to perform the program changes
4. Train the staff on project standards
5. Begin program conversion
6. Conduct unit testing
7. Set up the test environment
8. Create test data and revalidate the test plan
9. Conduct system testing. Maintain and resolve problem log and issues log. Maintain version control
10. Conduct regression test
11. Begin user and systems documentation
12. Monitor related projects, where systems are being replaced or upgraded
13. Complete project schedule and budget for implementation stage

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 project assessment report

ENDNOTES

SECTION I: THE YEAR 2000 DILEMMA

CHAPTER 1: IT'S BIGGER THAN A BREADBOX

[1] Holman W. Jenkins, Jr., "Turns Out the Year 2000 Problem is Just the Beginning," *The Wall Street Journal*, May 6, 1997, page A23.

[2] Rick 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.

[4] Amy Malloy, "Beat the Clock!," *Computerworld*, January 6, 1997, page 67.

[5] "Briefs: EDS Tests for 2000," *Computerworld*, June 9, 1997, page 67.

[6] Paul Strassmann, "Numbers Add Up to a Bigger Year 2000 Disaster," *Computerworld*, June 9, 1997, page 78.

[7] Capers Jones, "The Global Economic Impact of the Year 2000 Software Problem," *Software Productivity Research, Inc.*, January 23, 1997, page 45, 58.

[8] Steven Levy and Katie Hafner, "The Day the World Shuts Down," *Newsweek*, June 2, 1997, page 53.

[9] Rick Telberg, "Start Worrying: The Year 2000 Bug Is For Real," *Accounting Today*, June 2–15, 1997, page 6.

[10] Steven Levy and Katie Hafner, "The Day the World Shuts Down," *Newsweek*, June 2, 1997, page 53.

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

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

[2] Ibid.

[3] Holman W. Jenkins, Jr., "Turns Out the Year 2000 Problem Is Just the Beginning," *The Wall Street Journal*, May 6, 1997, page A23.

[4] Patrick L. Porter and Deborah Radcliff, "Time Bomb!," *Software Magazine*, March 1997, page 32.

[5] Ibid.

[6] Wendy S. Mead, "Chase Projects $250M Cost to Avert Year-2000 Glitches," *American Banker*, April 11, 1997.

[7] "Special Edition: The Year 2000," *Chase Week*, April 6, 1997.

[8] Thomas Hoffman, "Securities Group to Chief Execs: Get Involved," *Computerworld*, May 26, 1997, page 63.

[9] Bob Violino, "Ready for 2000?—Securities Firms Form Group to Plan for Year 2000 Compliance," *Information Week*, December 9, 1996, at http://www.techweb.com.

[10] Steven Levy and Katie Hafner, "The Day the World Shuts Down," *Newsweek*, June 2, 1997, page 53.

[11] Kim Girard and Robert L. Scheier, "Telcos Lag on Year 2000, Analysts Warn," *Computerworld*, November 11, 1996, page 8.

[12] Steven Levy and Katie Hafner, "The Day the World Shuts Down," *Newsweek*, June 2, 1997, page 53.

[13] Michael Mecham, "Industry Tackles Year 2000 Problem Systematically," *Aviation Week & Space Technology*, May 26, 1997, page 61.

[14] Stewart Deck, "Year 2000 Syndrome Strikes Again," *Computerworld*, July 22, 1996, page 8.

[15] Joe Gwinn, "The GPS Week 1024 Rollover," http://www.navcen.uscg.mil/gps/ggeninfo/gps 1999.txt or http://www.compinfo.co.uk/y2k/gps.htm.

[16] Gary H. Anthes, "Feds Face Year 2000 Crises," *Computerworld*, April 22, 1996, page 1.

[17] Committee 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.

[18] Joel 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.

CHAPTER 3: STRICTLY BUSINESS

[1] Chuck Mumm, Dallas Water Utilities, city of Dallas, from an interview with me in the summer of 1997. He was primarily referring to XT PCs.

[2] A white paper from Compaq Computer Corporation, found at http://www.wa.gov/dis/2000/survey/dt_hard/compaq.htm, dated January 1996.

[3] Ibid.

[4] Terry 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.

[5] A letter from Apple Computer, Inc. found at http://www.wa.gov/dis/2000/survey/dt_hard/apple.htm, dated July 8, 1996.

[6] A white paper from Compaq Computer Corporation, found at http://www.wa.gov/dis/2000/survey/dt_hard/compaq.htm, dated January 1996.

[7] Pete Woytovech, "The Century Rollover and the PC System Date," found at http://www.dell.com/support/tech/rm/r&dnews/vectors/3/v3_cent.htm.

[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.

[9] "What Does the Year 2000 Mean to Corel Users?" found at http://www.corel.com/2000.htm.

[10] "Lotus and the Year 2000," found at http://www.lotus.com/uk/220e.htm, March 26, 1997.

[11] "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.

[12] "What Does the Year 2000 Mean to Corel Users?" found at http://www.corel.com/2000.htm.

[13] "Lotus and the Year 2000," found at http://www.lotus.com/uk/220e.htm, March 26, 1997.

[14] Capers Jones, "The Global Economic Impact of the Year 2000 Software Problem," January 23, 1997, page 22.

[15] Ibid.

[16] Warren 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.

[17] William Farrell and Charles E. Phillips, "Year 2000 Watch: Survey at Year 2000 Conference," Morgan Stanley U.S. Investment Research, April 25, 1997.

[18] Ann 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.

[19] Kristi Essick, "IBM Rolls Out Several Net Support Products," *Computerworld*, June 2, 1997, page 56.

[20] Thomas Hoffman and Mitch Betts, "Year 2000 Glitches May Be Everywhere," *Computerworld*, May 19, 1997, page 1.

[21] Mitch Betts, "A Costly Specter," *Computerworld*, May 19, 1997, page 28.

[22] Ann 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.

[23] Mitch Betts, "A Costly Specter," *Computerworld*, May 19, 1997, page 28.

[24] Emmett Paige, Jr., "Statement before the Committee on Government Reform and Oversight," U.S. House of Representatives, April 16, 1996, http://www.army.mil/army-y2k/osd_Y2K/c3itest.htm.

[25] Robert L. Scheier, Gary H. Anthes, and Allan E. Alter, "When Mistakes Could Cost Lives," *Computerworld*, January 2, 1997, page 4.

[26] Ibid.

[27] "Federal Bank Regulators Outline Year 2000 Project Management Goals," FFIEC, http://www.ffiec.gov/y2k/y2kpress.htm, May 5, 1997.

CHAPTER 4: INSIDE AND OUTSIDE THE ORGANIZATION

[1] "What's News—Business and Finance," *The Wall Street Journal*, July 25, 1997, page 1, column 2.

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

[3] Michael Mecham, "Industry Tackles Year 2000 Problem Systematically," *Aviation Week & Space Technology*, May 26, 1997, page 61.

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

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

[6] Ibid.

CHAPTER 5: 2000 IS A LEAP YEAR

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

[2] Ibid.

[3] Ibid.

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

[5] John 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.

[2] Thomas 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

[1] Charles 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.

[3] Charles 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 17.

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

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

CHAPTER 8: ACCOUNTING ISSUES

[1] AICPA, *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.

CHAPTER 9: LEGAL ISSUES

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

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

[3] Ibid. Page 79.

[4] Warren 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.

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

[6] Ibid.

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

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

[9] Warren 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.

[10] Ibid. Page 13.

[11] Ibid. Page 16.

[12] Ibid. Page 20.

[13] Susan Adams, "The Bug Bar," *Forbes*, July 28, 1997, page 45.

[14] Ibid.

CHAPTER 10: INSURANCE ON YOUR BUSINESS?

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

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

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

[4] Ibid.

CHAPTER 11: HUMAN RESOURCES

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

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

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

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

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

CHAPTER 12: PROJECT MANAGEMENT ISSUES

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

[2] Robert L. Scheier, "Many Year 2000 Projects Still at Staring Gate," *Computerworld*, April 14, 1997, page 24.

[3] William Farrell and Charles E. Phillips, "Year 2000 Watch: Survey at Year 2000 Conference," *Morgan Stanley U.S. Investment Research*, April 25, 1997, page 4.

[4] Robert L. Scheier, "Year 2000 Scoreboard," *Computerworld*, December 9, 1996, page 85.

[5] Charles 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.

[6] Capers Jones, "The Global Economic Impact of the Year 2000 Software Problem," Software Productivity Research, Inc., January 23, 1997, page 41.

[7] Ibid. Page 18.

[8] Ibid. Pages 10, 11, 14.

[9] Ibid. Page 31.

CHAPTER 13: ENTERPRISE ISSUES

[1] Bureau of National Affairs (BNA) Daily Report for Executives, November 20, 1996, page A16.

[2] Capers Jones, "The Global Economic Impact of the Year 2000 Software Problem," Software Productivity Research, Inc., January 23, 1997, page 41.

[3] Scott Kirsner, "Millennial Blues: Preparing for Year 2000," Boston.com, http://www.boston.com/business/emerging/year2000.htm.

[4] Capers 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.

[8] The staff of *Computerworld, Computerwoche*, and IDG News Service, "Your Other Millennium Problem," *Computerworld*, May 26, 1997, page 70.

[9] Ibid.

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

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

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

CHAPTER 18: DATE MODIFICATION OPTIONS

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

CHAPTER 19: Y2K CONSULTANTS

[1] Glenn Cheney, "D&T Launches Software Solution for Y2k Blues," *Accounting Today,* July 28–August 10, 1997, page 16.

[2] Charles 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.

[3] Ibid.

[4] Ibid.

[5] Ibid.

[6] Ibid.

[7] Ibid.

[8] "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.itaa.org/yr2000bg.htm.

[9] Ibid.

CHAPTER 20: TOOLS

[1] Amy Malloy, "Beat the Clock!," *Computerworld,* January 6, 1997, page 67.

[2] C. Lawrence Meador, "Technology Overview: Solving the Year 2000 Problem," *Information Week,* February 5, 1996, http://www.techweb.cmp.com/iw/565/65olyr2.htm.

[3] Cathleen A. Gagne, "This Isn't a Sprint . . . It's a Year 2000 Marathon," *Computerworld,* March 31, 1997, page 85.

[4] Ibid.

[5] Amy Malloy, "Beat the Clock!," *Computerworld,* January 6, 1997, page 67.

[6] Thomas Hoffman, "Client/Server Users Must Push for Year 2000 Tools," *Computerworld,* June 23, 1997, page 43.

[7] Amy Malloy, "Beat the Clock!," *Computerworld,* January 6, 1997, page 67.

[8] Ibid.

[9] Anna Fortig, "The Path To Year 2000 Is a Tool Road," *The Year 2000 Solutions Series, Computerworld Custom Publications,* June 2, 1997, page 16.

[10] Robert L. Scheier and Thomas Hoffman, "Cashing in on Year 2000," *Computerworld,* July 14, 1997, page 59.

[11] Cathleen A. Gagne, "This Isn't a Sprint . . . It's a Year 2000 Marathon," *Computerworld,* March 31, 1997, page 85.

[12] Amy Malloy, "Beat the Clock!," *Computerworld,* January 6, 1997, page 67.

[13] Thomas Hoffman, "Client/Server Users Must Push For Year 2000 Tools," *Computerworld,* June 23, 1997, page 43.

[14] Amy Malloy, "Beat the Clock!" *Computerworld,* January 6, 1997, page 67.

[15] Cathleen A. Gagne, "This Isn't a Sprint . . . It's a Year 2000 Marathon," *Computerworld,* March 31, 1997, page 85.

[16] Amy Malloy, "Beat the Clock!," *Computerworld,* January 6, 1997, page 67.

[17] Anna Fortig, "The Path To Year 2000 Is a Tool Road," *The Year 2000 Solutions Series, Computerworld Custom Publications,* June 2, 1997, page 16.

[18] Amy Malloy, "Beat the Clock!," *Computerworld,* January 6, 1997, page 67.

[19] Capers Jones, "The Global Economic Impact of the Year 2000 Software Problem," Software Productivity Research, Inc., January 23, 1997, page 49.

[20] Anna 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.

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.

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.

disaster recovery. A backup method should primary computers fail to process company data, whether the failure is from fire, theft, weather, 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 28 years to avoid the year 2000 problem for another 30 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 on 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.

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.

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 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.

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

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 that 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***.

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 prompt medical care or they will die but if they get medical care they will have a good chance of living. The last group should receive the first priority medical attention.

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."

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

About the Author

Sandi Smith, CPA, CMA, CDP, is a Dallas-based sole proprietor who offers technology consulting and training to small- and medium-sized businesses, including CPA firms. She has 18 years experience as an accounting and systems professional with a mix of Fortune 100 and small company experience. She has an MBA in Management Information Systems. Sandi authored AICPA's *1997 Top Ten Technologies and Their Impact on CPAs* and has authored articles in *Management Accounting* and *Journal of Accountancy.* She enjoys receiving comments and questions at sandi@cyberramp.net and visits to her Web site at www.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—*InfoTech 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 Fax
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: ______________________________

E-mail 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 Schwartzberg, Specialized Publications & Subsidiary Rights, AICPA, Harborside Financial Center, 201 Plaza III, 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?

0 1 2 3 4 5
No knowledge Working knowledge Very knowledgeable

2. How useful is this publication to your line of work?

0 1 2 3 4 5
Not useful Somewhat useful Extremely useful

What is your current job title? ______________________________

3a. What did you like most about this publication?

3b. What did you like least about this publication?

4. Did the technical level of this publication meet your needs?

0 1 2 3 4 5
Too basic Just right Too advanced

5. Did this publication live up to your expectations? Yes ______ No ______

6. Where did you learn about the availability of 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

TOP TEN TECHNOLOGIES AND THEIR IMPACT ON CPAs

by Sandi Smith, CPA

In-depth coverage of the 10 most influential technologies, including applications, useful Web sites, leading vendors, and a list of other resources and reference materials, make this a valuable all-in-one resource for CPAs.

No. 042300KC $19.95/$24.95 Nonmember

THE CPA'S GUIDE TO WEB COMMERCE

by John Graves, CPA and Jacqueline Justice, Ph.D.

This guide covers various payment processing systems and how to implement them on a Web site. Use HTML to develop and display content and learn about security issues such as protecting a Web site. Plus—the optional Kent Web Design Toolkit features a CD-ROM and software to construct a Web site along with instructional videos.

Guide only: No. 093004KC Guide & CD-ROM: No. 093007KC
$29.00/$39.00 Nonmember $39.95/$47.95 Nonmember

THE CPA'S GUIDE TO INTRANETS

by John Graves, CPA and Jacqueline Justice, Ph.D.

Find out about Intranets and how they work, the software you need to get up an Intranet, how to use HTML to display Intranet Web pages, and how to maintain security. Plus—the optional Kent Web Design Toolkit features a CD-ROM and software to construct an Intranet Web site along with instructional videos.

Guide only: No. 093002KC Guide & CD-ROM: No. 093006KC
$29.00/$39.00 Nonmember $39.95/$47.95 Nonmember

THE CPA'S GUIDE TO INFORMATION SECURITY

by John Graves, CPA and Kim Hill Torrence

This guide explains how to assess security risks, select and lead the information security planning team, and develop and implement an information security plan.

No. 093003KC $29.00/$39.00 Nonmember

SOLVING THE YEAR 2000 DILEMMA

by Sandi Smith, CPA

You will learn how to recognize the serious problem facing you and your clients or employer as we get closer to January 1, 2000. *Solving the Year 2000 Dilemma* presents possible solutions as well as important practical issues including how to manage a year 2000 engagement.

No. 093008KC $29.00/$39.00 Nonmember

TALKING TECHNOLOGY

A monthly audio news magazine from Gartner Group

Every month, this unique subscription series delivers one audiocassette to help you stay current on hot topics like: IT business strategy . . . Process reengineering . . . Client/server technology . . . Database management . . . Document management . . . Electronic commerce . . . The Internet . . . Virtual reality . . . Year 2000 date change . . . and much more. **12 monthly Talking Technology audiocassettes.**

No. G03000KC $199.00/$499.00 Nonmember

BOOMER'S GUIDE TO CREATING A TECHNOLOGY PLAN & BUDGET

This product contains four distinct components: A step-by-step guide to help you prepare your technology plan, Boomer's Blueprint video to help you identify the key technology issues for your firm, The Technology Budget Calculator to prepare a three-year technology budget, and one hour of telephone consulting with Boomer staff.

No. 043011KC $495.00

BOOMER'S GUIDE TO CREATING A TECHNOLOGY BUDGET

This toolkit allows you to prepare a three-year technology budget. It provides a series of worksheets to help you gather data for your budget. The calculator is a proven budgeting tool that incorporates hardware and software components—you can even prepare technology budgets for your clients. Note: It is not necessary to purchase this *Guide* if you are purchasing *Boomer's Guide to Creating a Technology Plan & Budget.*

No. 043012KC $149.00

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. 090427KC $24.95/$31.00 Nonmember

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" with studies on a pension benefits boutique . . . how long-distance employees get a marketing edge . . . and much more!

No. 090426KC $22.95/$28.95 Nonmember

www.aicpa.org

093008