DANGEROUS DATES FOR SOFTWARE APPLICATIONS

Version 2 - March 24, 1998

Abstract

The year 2000 date problem is not the only calendar problem causing trouble for software applications. This article highlights some of the known date problems that are likely to affect software applications over the next 50 years. Other date problems that might impact software include the date at which global positioning satellites (GPS) roll over, the dates at which commodities switch to the Euro, the dates at which the UNIX and C libraries roll over, and some hazardous date patterns which have been used for non-date purposes in software applications. In addition, at some point early in the next century the numbers of digits assigned to social security numbers and telephones will run out of capacity.

Over the next 50 years at least 60,000,000 software applications will need modification because of various date problems. The total costs of these modifications can top $5 trillion dollars. The report concludes that because date problems with computers and software are so widespread, serious, and expensive, a new international standard for dates should be developed for computer purposes. The proposed date format includes a "key" field which is used to identify which specific date format follows. This method would allow older date formats to be used, and would support multiple calendars.

Capers Jones, Chairman
Software Productivity Research, Inc.
1 New England Executive Park
Burlington, MA 01803-5005

Phone	781 273-0140 X-102
FAX	781 273-5176
Email	Capers@SPR.com
CompuServe	75430,231
Web	http://www.spr.com

INTRODUCTION

Because the time it takes for the earth to move around the sun is 365 and roughly 1/4 days, keeping accurate time has been a problem for the human species ever since history began. Adding to the problem, the monthly cycles of the moon and the annual cycle of the sun are not exactly equal and only coincide every 19 years. Because the solar year is not evenly divisible by the number of days, calendars have needed periodic adjustments to bring them back into synchronization with the solar year.

The invention of atomic clocks and the advent of the coordinated universal time system (abbreviated UTC) have made the measurement of time the most accurate form of measurement known to the human species. While time at the level of seconds can be measured with an accuracy of 1 second in several million years, the coarser measures of dates and calendar intervals continue to be troublesome, and are particularly troublesome in the context of computers and software applications.

With the advent of computers and software, calendar and date problems have suddenly taken on a new importance. Now that so many important human activities are being performed by computers and software, date errors can disrupt interest rates, funds transfers, air traffic control, telephone systems, and electric power generation and cause a host of serious and often unexpected problems.

In theory computers and software should be able to keep track of time and dates with greater accuracy than any other human artifacts. Unfortunately the theoretical capabilities of computers and software for both and date and time keeping have not been achieved because of a historical stumbling block: Storage costs for date information were high enough so that insufficient space was allotted for full precision. Therefore whenever possible, storage space was conserved.

Two common methods of recording dates are used in computers and software applications: 1) Storing dates using conventional date representation formats such as the year, month, and day; 2) Accumulating the number of seconds from an arbitrary starting point. Both of these methods have been troubled by insufficient storage space.

For conventional date formats the most traditional way to conserve space was to truncate four-digit years fields and use only two digits so that a year such as 1998 would be stored as 98.

For the method of keeping track of dates by accumulating seconds from an arbitrary starting point the size of the field is usually limited to no more than 4 bytes of storage space. This method will "overflow" and reset to zero or to the initial starting point when capacity is exceeded.

For example, the date "buckets" used on global positioning satellites (GPS) record time for 1023 weeks, and then reset to week 0 on the 1024th week and continue this cycle about every 20 years.

Starting in August of 1999 and continuing at intervals over the next 50 years both methods used by computers and software for dealing with dates will experience problems because of the historical practice of attempting to conserve storage space.

Both of these methods have worked reasonably well up until now, but both will run into serious problems when their storage boundaries are exceeded. What happens when the computer date storage volumes are exceeded is now a very serious issue which can cause untold economic damage and perhaps physical damages too, in the sense of shutting down electric power plants, stopping assembly lines, or grounding aircraft. Let us examine some of the known date problems that are going to affect computers and software over the next 50 year period.

Several other problems associated with insufficient numbers of digits will occur during the next 50 years too, but at moments in time that are somewhat unpredictable in 1998. An interesting report by Dr. Clifford Kurtzman (Kurtzman 1997) notes that the population of the United States will exceed the capacity of 7-digit phone numbers around the year 2025. We are already experiencing frequent problems with the need to reassign area codes. The capacity of U.S. social security numbers (9 digits) will be exceeded by about the middle of the century, say 2050.

The year 2000 is a specific instance of a general problem which will trigger massive expenditures unless it is solved once an for all. The general problem is the assignment of an insufficient number of digits for key numerical information. This problem already manifested itself circa 1980 when some applications had to be modified because they did not have enough digits to keep pace with salaries and overtime when compensation levels began to top $100,000 for many professions.

Between now and roughly the year 2050, a huge amount of effort and hundreds of billions of dollars of costs will spent on expanding numeric fields in software applications:

- Financial fields starting circa 1980
- Zip codes starting circa 1985
- Date fields starting circa 1999
- Telephone numbers starting circa 2025
- Social security numbers starting circa 2050

The cumulative costs of expanding numeric fields as their capacity is exceeded will erode many of the economic advantages of the use of computers and software. It is obvious that a more permanent general schema must be developed before the maintenance expenses trigger bankruptcy and litigation for hundreds of corporations and even for some governments.

Incompatibilities Among International Date Formats

For centuries the way dates are represented when they are printed have varied from country to country. These variations presented no real problem until the advent of the computer era. Even with computers, the problems were fairly minor but it was obviously necessary to know which date format was used to ensure correct date calculations.

For example, in the United States we normally use a format of "month, day, year" such as

10/6/98 for October 6, 1998. In much of Europe the same date would be printed using the format of "day, month, year" or 6/10/98 for the same day. Obviously the European form might be misinterpreted as June 10th in the United States, or the U.S. format might be misinterpreted as June 10th in Europe if the software assumed the wrong alternative.

To facilitate international trade and commerce using computers and software, the International Organization of Standards (abbreviated to ISO) has proposed a standard date format that expands the number of year digits from two to four. This is the well-known ISO standard 8601: 1988(E). This same format is supported by the American National Standards Institute (ANSI) and also by the National Institute of Standards and Technology (NIST).

The ISO date format puts the year first, then the month, and then the day using the format yyyy/mm/dd. Thus the date of October 6, 1998 would be represented as 1998/10/06 using the proposed ISO standard. (Note that the slash symbols "/" are not part of the date standard but are simply used here to enhance legibility on the printed page.)

Unfortunately, the most common date format used in the United States works in the opposite direction, and puts the years last. This is the default representation on various Microsoft products, although Microsoft's products can support the ISO format too. Thus for many personal computer applications in the United States, dates are represented in the sequence of the month first, followed by the day, followed by the year: mm/dd/yyyy. Thus October 6, 1998 would be 10/06/1998 using Microsoft's U.S. default date format.

Unfortunately the four-digit ISO standard for date formats is not fully adequate. Both the ISO standard and the normal U.S. date representation share a common failing when trying to deal with dates and computers. Both of these date formats exhibit unconscious attempts to conserve storage space without realizing that this is causing unnecessary problems.

By adding at least one extra digit to the ISO date format, any date representation could be accommodated by using the extra digit as a key (shown as "x" in the examples) to identify whether the ISO date format (x-yyyy-mm-dd) or the U.S. default date format (x-mm-dd-yyyy) was intended. The key could also identify other alternatives, such as the normal European date format (x-dd-mm-yyyy) or even Julian dates, which record the number of days from the beginning of a year starting with 1 and running to 365 or 366. Even the traditional Japanese dates based on Imperial reigns could be accommodated.

Using an extra digit (or digits) as a key with the following meanings would make identifying which date format is intended a lot less messy than the current situation. Today ascertaining which of the many possible date formats might be used in software applications either requires advance notification to programmers and users, or extraordinarily complicated algorithms for deriving dates, with no absolute way of knowing if the date format selected is the right one without inspection or testing. Consider how versatile date logic would be if one or more extra digits were utilized:

Possible Date Format Key Using One Additional Digit

Key	Definition
1	ISO date format with four digits for years (yyyy-mm-dd)
2	U.S. default date format with four digits for years (mm-dd-yyyy)
3	Normal European date format with two digits for years (dd-mm-yy)
4	Normal European date format with four digits for years (dd-mm-yyyy)
5	Normal U.S. date format with two digits for years (mm-dd-yy)
6	Julian date with two year digits (yy-ddd)
7	Julian date with four year digits (yyyy-ddd)
8	Astronomical time (ddddddd) starting from January 1, 4713 BC

The example shown above illustrates what might be done using only a single extra digit. For many date and time-keeping purposes, it might be desirable to include not only century, year, month, and day information but also weeks, hours, minutes, and seconds. Thus if a date key is used to identify which format is being utilized, even the following 16-digit date format could be used if needed:

x-yyyy-MM-ww-dd-hh-mm-ss

In this 16-digit format x is the date code; yyyy represents years; MM represents months; ww represents weeks; dd represents days; hh represents hours; mm represents minutes; and ss represents seconds. Even 16 digits is not enough precision for some uses, so the schema could be extended down to the nanosecond level. If it takes 20 digits or more, but any known date format might be incorporated into the schema, then conservation of space is irrelevant.

For a universal date format there may be hundreds or even thousands of date variants which would need specific keys. Therefore a 4-digit key following by 20 digits of date information should be able to accommodate any known calendar, and operate over arbitrarily long time periods.

Incidentally, the ISO standard date format is not adequate for scientific purposes. For dealing with geological time periods, spans of millions of years must be accommodated and most of this time would be in the BC era and hence require negative numbers. For astronomical time, billions of years must be accommodated. Indeed, for astronomical purposes the calendars of other planets such as Mars may eventually need to be accommodated.

The main feature of the author's proposed date format is that it is aimed at storing dates in computers in an era where unlimited optical storage is the rule. Therefore it is unwise to continue to develop date formats whose utility is compromised by an unconscious need to conserve storage space. Let us design a computerized date format that can last indefinitely, support scientific as well as business dates and time, and support all of the older date format variants. As the situation now stands, there are no current or proposed

date standards by ISO or anyone else that are fully adequate even for business if it is transacted by computers, to say nothing of scientific purposes.

Under current date formats, it is almost impossible to utilize technologies such as data mining and on-line analytical processing (OLAP) for scientific data associated with geology, archeology, astronomy, etc. because the dates involved exceed the ranges of standard date formats, and in many cases, exceed the date handling ability of normal business software applications such as spreadsheets and data base packages.

Adding extra "key" digits to date formats when used in computers would allow any conceivable date format to be included in the general schema, so that geologic and astronomical time, Julian dates, the Chinese calendar, the Jewish calendar, or even the Aztec calendar could be utilized as needed.

The date key would not have to be printed or appear on screen, but the presence of a date key would enable software applications to handle calendar calculations with far greater ease and flexibility than has ever been possible since computers became business and scientific tools.

It should be noted that the general solution of using a key field to identify which specific numeric or alphanumeric format follows can be used to deal with other problems besides dates. This same method might be used to handle the international variations in zip code formats, or the international variations used for social security numbers, or their equivalent in other countries.

Because so many messy date and numeric field-length problems for computers and software are going to occur over the next 50 years, it would be highly desirable if we could create a long-range standard that enabled us to handle dates in computers that supported all calendars and all date representations. We can never accomplish this unless we get over the unconscious need to conserve storage. Storage is now cheap, and if assigning 20 digits can simplify date logic and date calculations, now is the time to do it.

An expanded date format would require changes to software applications and databases and would be expensive to implement. But between the year 2000 problem the UNIX date rollover and other date problems, we are already going to spend several trillion dollars in software date changes . Right now, the new replacement dates will have the same kind of problem as the current dates: They don't have enough digits to handle scientific date purposes, they too will overflow, and eventually they will have to be changed again.

Make no mistake about it:, computer and software dates and numeric data are far more important than printed dates. It would be enormously valuable if a truly effective date standard could be developed. Right now none of the current date standard formats are going to accomplish anything but cause more long-range problems for software and computer vendors, and a continuing need for tricky and error-prone date calculations.

The author recommends that an international symposium be held on computer date problems which would deal not only with business date recording but also with the need for recording scientific dates and other numeric data such as telephone numbers. The

purpose of this symposium would be to develop a date strategy which would eliminate the kinds of problems which we are about to experience between 1999 and 2038 when the Euro kicks in, the GPS date roll-over occurs, the year 2000 event happens, and the UNIX date roll-over occurs in 2038. Let us now consider some of the major date problems that the computer and software industry will be facing over the next 50 years.

January 1, 1999 (The Beginning of the Euro Currency era)

The European Union is moving toward a unified currency (the Euro) which is scheduled to be introduced during the period starting in January 1, 1999 and running through calendar year 2002. There are two significant date problems associated with the Euro:

1. The timing of the Euro introduction.
2. The impact of the Euro cutover on software and data mining.

The timing of the Euro is one of the worst public policy decisions in human history because it pits the world's second largest software project (the Euro) against the world's largest software project (the year 2000). There are not enough software personnel available to complete either one of these massive efforts in time, and the whole idea of trying to accomplish both of these on the approximately the same schedule is going to cause major economic problems.

For additional information, refer to the author's book "The Year 2000 Software Problem - Quantifying the Costs and Assessing the Consequences" (Addison Wesley Longman, 1997). See also the author's short articles on "Resource Conflicts Between the Year 2000 and Euro Currency Software Problems" (*Year 2000 Journal*, January/February 1998) and "Rules of Thumb for Year 2000 and Euro-Currency Software Repairs" (SPR technical report, February 1998).

The timing of the Euro is going to teach politicians a very painful lesson: they are no longer in control of many national events. In the pre-computer era when political directives were implemented by human beings, politicians could set arbitrary dates and expect to have their decrees implemented more or less when they wished.

In the computer era, political decisions which trigger massive software updates can no longer be scheduled using arbitrary end points determined by political processes, treaties, or government decrees. If massive software updates are involved, then the timing must be derived from the ability to accomplish software and database updates. Thus the Euro will not be implemented as planned no matter what politicians say or think, because the necessary software updates will not be ready in time.

Over and above bad timing, the software implications of the Euro are horrendous and may cause unexpected problems with stock market trading, banking, and other financial software applications. In addition, the Euro may interfere with data mining, data warehousing, on-line analytical processing (OLAP) and all other forms of analysis which look for trends over time.

The harmful date implications of the Euro are due to the fact that thousands of

commodities such as stocks, bonds, manufactured products, services, etc. will begin to switch from local currencies to the Euro when the Euro rolls out.

Many software applications can handle the appearance of new currencies since they have occurred from time to time throughout history. The problem with the Euro is that no software applications were prepared to deal with the situation that products which have been on the market for many years will abruptly start being priced using the Euro rather than their present currencies in the same countries.

This means that any software application which does long-range trend analysis before and after the Euro introduction will have to include complex currency conversion algorithms. Either the older cost data will have to be converted to the Euro, or the newer Euro costs will have to be backfitted to the older currency. Sometimes it may be necessary to do both: show prices in terms of both the Euro and the old currency.

Suppose you have been following the long-range stock prices of a European company such as Siemens-Nixdorf. After the Euro is introduced, the stock values will be displayed in Euro's rather than in Deutschmarks, so all of your long-range trend analysis must handle the currency conversion as of the date when the stock makes the transition to the Euro.

Suppose you are interested in long-range trends concerning the price of a basic commodity such as the cost of wheat in Germany. The introduction of the Euro will cause an abrupt discontinuity is historical data analysis, and all long-range trend analysis will have to bridge the pre-Euro and post-Euro changeover.

Thus many thousands of software applications must be modified to handle the change from older currencies to the Euro. Also, data bases and data warehouse and data mining operations must also deal with the currency changeover.

Making the situation still more difficult, the Euro is being introduced over a sliding time period which will vary by country, industry, and commodity. Thus the dates for the conversion of specific products and services from local currencies to the Euro will take place at almost random intervals between 1999 and 2002, or longer since it is obvious that the Euro is going to run late.

The changes needed to fully support all of the implications of the Euro are not trivial. It is easy enough to add a new currency to software applications, but to support historical analysis of products and services which have been marketed for 50 years in one currency and abruptly switch to being marketed in another currency is a very serious issue for long-range trend analysis.

As can be imagined, the Euro is going to cause major problems for data mining, on-line analytical processing (OLAP) and any other form of long-range trend analysis which spans the pre-Euro and post-Euro periods.

The politicians of the European Union appear to be blissfully unconcerned about the damages which the Euro will cause to software applications and business operations, and about the huge expenses needed to recover from the Euro introduction.

Indeed, as of 1998 many European politicians are still naively saying that the Euro will be introduced on schedule when it is painfully obvious that politicians have failed to estimate the schedules and costs of the necessary software updates. So far as can be determined, neither the European Union nor any of the national governments involved have produced a reasonable cost, schedule, quality, risk, and damage estimate for the Euro introduction. Indeed, the very real risks and damages have been ignored by most European political leaders.

The Euro may commence on schedule in 1999, but it will not be completed by 2002 regardless of assurances by European politicians. As the Euro begins to enter circulation and financial software lurches into the Euro era, we can anticipate unexpected failures and delays in many business operations. We can also anticipate several forms of litigation, including but not limited to the following:

1. Lawsuits against the European Union for damages and cost recovery
2. Lawsuits against national governments for damages and cost recovery
3. Lawsuits against companies whose Euro updates are imperfect and damage clients
4. Lawsuits against corporate officers by shareholders

It is an interesting question to ask if anyone has performed a cost/value analysis of the Euro. There may be some substantial long-range economic benefits from a unified European currency, but the ROI from 1999 through about 2005 appear to be distressingly negative. Indeed, the Euro software update costs will be so high that it is possible that the long-range value of the Euro may not exceed the damage costs until perhaps the year 2020 or 2025 if even then.

Now that computers and software are the dominant tools of commerce, industry, and business arbitrary government decrees such as the Euro which trigger massive software updates can be viewed as a form of hidden taxation. At the very least, private business is being forced to act as an unpaid agent of national governments.

In the future it is likely that government mandates such as the Euro which trigger immense business costs should include some form of relief for the affected companies, either in the form of reduced taxes, subsidies, or some other method of compensation for the work involved in software modifications.

Incidentally there is a minor but annoying software problem which will affect many thousands of software applications which display currencies, even if they don't do any complicated form of currency calculation. As this is written in 1998 there is no easy way to display the Euro currency symbol on a computer screen or print it out!

Unfortunately the new Euro symbol is not part of many older character sets. Under Windows 95 it is very difficult to print the Euro symbol, and currently only Microsoft Word can accomplish this without excessive difficulty. Under Windows NT and presumably under Window 98 the symbol will be included and easier to access.

Other computer hardware and software platforms are also not quite ready for printing or displaying the Euro, since the symbol has only started being included circa 1996. This

means that many older computers, or older software applications, will not be able to display the Euro currency symbol even if they can handle the currency conversion calculations.

GPS End of Week Rollover (August 21/22 1999)

The current network of 24 global positioning satellites (GPS) keeps track of dates by recording the number of weeks from midnight on January 5th 1980 using a modulo 1024 approach. That is, the week counter will reset to 0000 after week 1023, which occurs at midnight on August 21, 1999. The GPS dates will of course roll over or reset to 0000 every 1023 weeks from then on, or about every 20 years so long as this method is utilized.

The roll-over of GPS dates has been clearly documented in the original standards (ICD-GPS-200), and all of the major GPS receivers and ground support units should have planned for the roll over and be ready to deal with the situation. Therefore on the face of it, the GPS rollover should not be a major technical problem.

However, we are dealing with applications that were designed and coded by human analysts and by human programmers. Because humans are involved, it is very likely that the GPS end of week 1023 rollover may have been missed or not handled properly in some ground stations or in some software applications which use the GPS system. Therefore many companies and government agencies which use the GPS system are now starting to perform tests of ground units and software applications to ensure that the roll-over will be handled correctly.

For navigational purposes, missing the roll-over may or may not have a major impact: probably not. But the GPS system is also used for purposes where a missed roll-over can have a major impact indeed. The GPS dates are used to synchronize major international funds transfers to ensure that interest payments are calculated to the second. For these applications, an error in missing the end of week roll-over can be very troublesome indeed.

Unfortunately, the GPS roll-over date of August of 1999 is putting this possibly minor date problem in the path of the much more serious year 2000 problem, which occurs only a few months later. This means that applications which use GPS time signals must be tested for both the GPS roll-over and the year 2000 problem more or less simultaneously.

The March 1998 issue of the journal *IEEE Spectrum* contains a useful article by David Allen, Neil Ashby, and Cliff Hodge which gives an overview of the GPS time-keeping methodology and equipment.

Because time kept by atomic clocks is not exactly the same as the solar time of the earth around the sun, there are small differences between UTC and GPS times The UTC utilized leap second to more or less match time measured atomically with the somewhat less precise period of the earth's rotation. The GPS time does not include leap seconds. Eventually these differences will have to be dealt with.

The "Nines" End of File Date Problems (September 9, 1999 and September 8, 2001)

Software applications need some way of indicating the ends of files, so that they can be shut down safely. One technique used for this has been to use a number such as 9999 as a file termination code. The problem with this method is that although 9999 is not intended to be a normal date, it might well be misinterpreted by some software applications and be considered as the date September 9, 1999. If the 9999 sequence is misinterpreted as a normal date, then obviously calculations will be in error.

A similar problem has been noted in UNIX applications where 999,999,999 has sometimes been used to indicate an end of a file situation. As it happens, the number string of 999,999,999 under UNIX is actually the date of September 8, 2001.

The implications of this are that although the normal expiration of the UNIX clock is not until 2038 AD, some applications written in the C programming language and running under UNIX may experience date problems in the Autumn of 2001, or 37 years before the main UNIX date problems occur.

Thus the aphorism that "UNIX does not have a year 2000 problem" may be true, but UNIX may well have some year 2001 problems in specific applications. UNIX does have a major 2038 date problem that will prove perhaps as troublesome as the year 2000 problem is proving to be.

The root cause of the problem is using end of file symbology which is potentially ambiguous. It would be useful to define a standard end of file representation that worked under all operating systems and on all hardware platforms and would not be misinterpreted as a date.

What comes to mind as a possibility for this might be using the infinity symbol "(" rather than a numeric sequence, although another possibility would be to create a specific end of file symbol which was used only for that purpose, and hence would have no possible ambiguity at all. Whatever solution is adopted, the problem of creating an unambiguous method to indicate file termination points is a problem that should be solved once and for all.

Although this problem has been solved on many operating systems, it is not clear how many older applications still use strings of nines with ambiguous meanings.

The Year 2000 Date Problem (December 31, 1999)

Many articles have been written on why the year 2000 problem will occur, so it is only necessary to include a short background discussion here. The root cause can be traced back to the early days of computers, when information was stored on punched cards or paper tape, or on magnetic tapes and the early disk drives used for mainframe computers.

Data storage was so limited and so expensive that any method that could save storage

was readily adopted. Since no one in the 1950's or 1960's had any idea how long software would last, it seemed natural to store dates in two-digit form; i.e. 1965 would be stored simply as 65. This method was convenient and seemingly effective.

It is unfair to blame computer programmers for the year 2000 problem, even though it is the presence of the problem in computer programs that is troublesome. The year 2000 problem actually originated as an explicit requirement by clients of custom software applications and the executives responsible for data centers as a proven and seemingly effective way of saving money.

When the current two-digit date requirement actually became a U.S. military and government standard, programmers who knew that problems might occur were constrained by the standards to use the two-digit form. It is plainly not very effective to raise concerns about problems that won't occur until long after all of the executives and clients who might repair the situation have retired.

By the late 1970's and early 1980's it started to be noted that software applications were sometimes having remarkably long lives. For example, IBM's MVS operating system was approaching 20 years of age, as were a number of other widely used applications. Some tremors of alarm about date limits began to show up, but there was still no immediate serious alarm since the end of the century was 20 years away.

Some applications began their date field expansions in the 1970's, such as mortgage companies who had to deal with 30 year mortgages. Also, many insurance companies always utilized four digit year fields since their actuarial tables spanned more than a hundred years.

Unfortunately applications which continued to use two-digit years outnumbered applications which used four-digit years by roughly a ratio 20 to 1. The bulk of the two-digit years were in software applications which used dates for immediate purposes, rather than long-range trend projections. Thus two-digit dates were used in operating systems, embedded applications, and many government software applications as well as business software.

The year 2000 date problem began to attract significant public and professional notice in 1995 when credit cards with five-year expiration dates hit the year 2000 barrier and stopped being accepted. The temporary solution adopted was to cut back the expiration dates from five years to three years to allow credit card companies time to make repairs.

From that point on, year 2000 problems have occurred with increasing frequency as applications which look forward begin to run into the year 2000 barrier. The worst problems will occur at or near the end of the century when electric power plants, telephone networks, and specialized equipment with embedded software begin to encounter hidden date problems and malfunction. There will also be problems occurring before the actual end of the century for companies and government agencies whose fiscal years are decoupled from the calendar year.

There are a number of strategies for fixing the year 2000 problem. But it is now 1998 and there is no longer time available for changing date fields in software and data bases from

two-digit to four-digit formats. A variety of "masking" approaches are being used which do not actually change the date format, but utilize external, outboard software tools to convert dates entering and leaving the application. Following are brief discussions of major year 2000 repair strategies:

Date field expansion: Expanding date fields from two-digit to four-digit form is the "classic" method for solving the year 2000 problem. This method provides a permanent solution, but has proven to be costly and difficult for many applications. Sometimes dates are indirect and hidden. For example, dates may be embedded in product serial numbers such as "1234**98**4321" where digits five and six show the year 1998 embedded in a 10 digit serial number. If date field expansion is the primary method used, it takes between 36 and 40 calendar months to completely convert all of the dates for a major company, so unless you started in 1996 there is no longer time to use this approach except for a few key applications.

Windowing: The windowing method establishes a fixed interval time period, such as 1915 to 2014, and uses external program logic to deal with all dates within that period. Assume that your window runs from 1915 to 2014. Dates below the mid point of the window such as 03 or 10 are assigned to the 21st century as 2003 or 2010, while dates above the mid point such as 97 or 98 are assigned to the 20th century as 1997 and 1998. Windowing for a portfolio can be finished in roughly 18 calendar months, so this has become one of the popular methods with late starters. However, windowing exacts a performance penalty and assumes that everyone using the data or the application knows about the existence of the windowing routines.

Compression: It is obviously possible to use some form of encoding within the allotted two digit date space to represent any conceivable date. By using a binary or hexadecimal representation rather than a decimal representation, the available two digits date field can handle dates over almost any period. Here too the work could be accomplished in less than two years. However, compression requires knowledge of the specific compression technique used by all applications accessing the data. There will also be performance reductions, but not as severe as windowing.

Encapsulation: This method uses an external tool and simply shifts all dates downward by 28 years, so that the year 2000 would be represented as 1972. The rationale for using 28 years is that a 28-year shift will bring the days of the week (i.e. Monday, Tuesday, Wednesday, etc.) and the calendar dates (i.e. October 6, 7, 8, etc.) into correct synchronization. The encapsulation method has the advantages of being fairly easy to do and can be finished before the end of the century. However, here too there is a performance penalty. Also some dates are subtle and calculated by indirect means such as dates hidden in serial numbers.

Bridging: This is a hybrid method used for data-base applications where the software itself is converted from two-digit to four-digit form, but the underlying data base is not due to the excessive difficulties associated with data base date field expansion. A fixed or sliding window or encapsulation are used with the data base itself. Here too, a performance penalty is exacted. Bridging is also used among late starters because of the chance of finishing in less than two years.

Data Duplexing is a specialized method for dealing with data base year 2000 problems, without changing the date fields of all of the applications which reference the data. Two versions of a data base are created, with one version containing the original two-digit date fields, and the second or "cloned" version containing the same information, but the date fields are expanded to four-digit form. Unfortunately doing this for a large portfolio is about a 36 month undertaking, so the optimal time for data duplexing has expired.

Obviously data duplexing requires a lot of work in keeping both versions of the data base synchronized. Data duplexing is a rather complex and expensive strategy, but actually expanding the date fields in data bases is one of the most troublesome and expensive aspects of the year 2000 problem.

Object-code date interception: Experimental methods for intercepting dates in executable object code are being researched and are now entering the market. The object-code interception method is just being demonstrated in IBM mainframe environments, but does not yet have commercial tools for other platforms. However, object-code date interception only works for explicit dates, and not for hidden or obscure dates such as the example of a date embedded in a product serial number. This method is the quickest, of course, and might be deployed in less than a calendar year. It is the last hope of the laggards, but it will probably not turn out to be a "silver bullet" since so many obscure dates would be missed.

Other alternatives for dealing with the year 2000 problem are also running out of time. Replacement with commercial packages is an option, but this approach does not work for custom software. It is far too late to build major replacement applications. Therefore, now is the time to start contingency planning on how to deal with date problems that won't be fixed in time.

Incidentally, in the entire 50 years of the software industry there has almost never been a major software application released to users where 100% of the latent errors were found prior to deployment. The current U.S. average overall is about 85% of defects are removed and 15% get deployed.

There are roughly 36,000,000 applications running in the world which have year 2000 date problems in them. It is very naïve to think that 100% of these will be repaired in time. It is also naïve to think that for any specific application that 100% of the year 2000 date references will be found and repaired.

The "best in class" removal efficiency for coding errors is less than 99% circa 1998, and the average is below 95%. While year 2000 specialists with automated search engines might achieve 99.9% defect removal efficiency, there is no reason to believe that ordinary programmers will exceed historical average results of roughly 95%. There will be year 2000 problems present at the end of the century, and hence contingency planning is needed right now.

The Year 2000 Leap Year Problem (February 29, 2000)

The 365-1/4 day rotation of the earth around the sun means that it is not possible to develop a calendar with a fixed number of days. Roughly every four years another day has to be added. Of course the situation is really much more complex because the rotational difference is slightly more than a quarter of a day, so just adding one day every four years only works for a century or two.

There are three general rules for determining a leap year, but one of these rules is so rare that it does not often occur and few people understand it. Because of this third rule, the year 2000 is a leap year and this aspect of the year 2000 problem will cause trouble on February 29th, 2000 AD.

Rule 1: Years divisible by 4 are leap years.
Rule 2: Years divisible by 100 are **not** leap years.
Rule 3: Years divisible by 400 are leap years.

Thus the year 2000 is going to be a leap year based on rules 1 and 3. It would not be a leap year based on rule 2, but the year 2000 is one of those rare years where it is necessary to account for the fact that the solar year is not exactly 365-1/4 days but slightly longer.

The implications of missing a leap year can be quite disruptive of computerized software applications. The year 1988 was a leap year which was accidentally omitted by a software vendor of mainframe security systems. Starting at midnight on the 28th of February 1988, customers began to be locked out from their computers because February 29th was not considered to be a valid date. By the time the company opened in the morning hundreds of frantic telephone calls and faxes were arriving from clients all over the world.

The failure mode of missing a leap year is either to shut the application down completely, or to cause calculations to be double posted. In any case, this problem is quite troubling and needs to be dealt with.

Because the year 2000 leap year not a normal leap year but one determined by the "400 rule" it will probably be missed by more than a few software applications. Unfortunately the much more visible year 2000 problem has obscured the leap-year problem but it will be troublesome too.

As of 1998 the year 2000 leap year status is probably one of the best known in history. The problem is that when software applications were being constructed in the 1970's and 1980's, the fact that year 2000 is a leap year often escaped notice. Therefore many legacy applications may fail in February of 2000 even if they make it past December of 1999.

The 10-Digit Telephone Number Problem (Circa 2025)

The author's business telephone area code changed from 617 to 781 in January of 1998. This change of course necessitated reordering business cards, office stationary, and all other documents and brochures which contained our phone number.

The change also necessitated notifying all of our customers, suppliers, etc. At least 300 companies had to be notified, and this area code change probably affected at least 2000 data bases for the author's company alone. It is hard to reach clients and suppliers with 100% efficiency, and indeed a magazine attempted to fax page proofs to our old area code. Because it was a fax instead of a voice line, the operator did not hear the recorded message about the area code change and the page proofs did not arrive.

This incident is a foretaste of more serious problems which may occur circa 2025 when the number of telephone numbers begins to exceed the overall capacity of the number of digits available. In the United States three digits are assigned to geographic area codes, and seven digits are currently assigned to the telephone number itself.

By the first quarter of the next century, we will begin to exceed the capacities of both the three digit area code fields and the seven digit telephone number fields. What may be needed for long-range stability might be five-digit area codes and perhaps nine digits for telephone numbers.

When the saturation point for telephone numbers is reached, massive software update costs will be needed. Also, millions of hand-held personal information managers will become obsolete because they cannot handle expanded formats for telephone information.

This problem will not shut down computers and damage national infrastructures as will the year 2000 problem, but it will still trigger billions of dollars in software upgrade expenses and will make telephone communication less certain than desirable.

Other countries besides the United States are also approaching the need to expand both area codes and add digits to basic telephone numbers as populations grow and new businesses are created.

The UNIX and C Library Problem (January 19, 2038)

As another example of field size causing date problems, on January 18th of the year 2038 yet another date crisis will occur when the UNIX operating system and the C programming language internal date representations expire. UNIX stores dates in terms of the number of seconds accrued after January 1, 1970 using a four-byte storage area.

Using normal 32 bit storage this method works until UNIX time reaches 2,147,483,647 accumulated seconds, when a roll-over occurs. Thus the UNIX clock will roll-over on January 19, 2038 at 3:14:07 at which point it will seem to be 1970 again or at least the number of seconds from January 1, 1970 will seem to be 0.

Some applications may then revert to January 1, 1970 as the current date, but some may revert to a date of December 13, 1901 based on implementation logic.

Here too, the root cause of the problem is conservation of storage by using only four digits. Using six digits instead of four would have extended the useful UNIX and C date

life for many thousands of year, but the use of four digits will cause another mass migration of dates in less than 40 years.

The C runtime library has a time function which reports time as a 31 bit signed integer. Jan Huffman of Software Productivity Research (SPR) has suggested creating a new data type which would be an unsigned integer, and hence allow the 32nd bit to be used for dates. This would provide an additional 68 years before roll-over occurs.

As of 1998 almost no press coverage is being given to the UNIX and C library problem because it is about 40 years away, even though this problem will be in the same magnitude as the current year 2000 problem.

If the UNIX problem follows the same pattern as the year 2000 problem, it will not be covered in the press until about 2033 and major repairs won't begin until 2036, when it is almost too late to get the affected applications updated before the roll-over occurs.

Date Expiration in Microsoft Products (2019 - 2078)

Because software from Microsoft is used in more computers than all other vendors combined, how Microsoft handles dates is a very serious issue. Based on Microsoft's congressional testimony in 1996 and the Microsoft year 2000 web site, information on all Microsoft products is available for review and analysis. Readers can start at the basic Microsoft web site and branch to relevant sections: http://www.Microsoft.com is the basic URL to get started.

Microsoft's internal standard is to record dates using four digits for the year, regardless of how the dates are displayed on screen. Users can select a variety of screen representation methods based on their own preferences. However if they select two-digit year formats then users are responsible for any problems this might cause for dates that run into the next century.

Although Microsoft states that all of their software is year 2000 compliant, there are still a number of date expirations which users should know about. For example the very popular Excel 95 spreadsheet package handles dates only up to 2019 using two-digit dates, or up to 2078 using four-digit dates. The Microsoft Project planning tool can handle dates only up to 2049. The Microsoft Access database product will stop at 1999 using two-digit date formats, but goes all the way to 9999 using four-digit dates.

According to Microsoft's congressional testimony, these rather close-in dates are going to be stretched out in future versions with the year 9999 being the stated end point for many new Microsoft product releases.

Microsoft was somewhat late in realizing the seriousness of the year 2000 problem, but is finally alert to its significance. However it would be useful to the scientific community if Microsoft adopted a more far-reaching date format which could deal with geologic and astronomical date processing.

The Social Security Number Problem (Circa 2050)

The use of unique national identity numbers such as the social security number in the United States poses a very difficult long-range challenge for software applications. The reason for the problem is that unlike many other numbers (i.e. telephone numbers) the social security numbers are "retired" once they are used and cannot be reassigned. Currently even social security numbers assigned to people who have died cannot be reassigned.

The current format for the U.S. social security number is nine digits long in the format nnn-nn-nnnn. This format uses the first three digits for state identification. The capacity of the social security number system is about 1 billion unique numbers. A report by Dr. Clifford Kurtzman (Kurtzman 1997) indicates that about 383,000,000 social security numbers have been assigned to date, and about 6,000,000 more are being assigned each year. Thus in theory the number of digits should serve for another 75 years.

But consequences of losing the integrity of the social security numbering system are very profound and will affect all forms of financial and government applications. Therefore it would be folly to wait until the last minute before taking remedial action, as we have done with the year 2000 and Euro-currency problems. Therefore the author recommends a target date of 2050 for creating an expanded schema for handling social security numbers.

It is obvious that some form of universal personal identity code is in the wind. When planning the expansion or replacement of social security numbers it is obvious that other issues need to be addressed. It would be a serious mistake just to add one more digit, when a more thoughtful solution is needed.

Whatever the solution adopted, a change involving social security numbers will have a major affect on millions of software applications and hence trigger expenses of billions of dollars.

Incidentally, although the U.S. social security number is cited in this article, the same form of problem occurs in virtually every nation in the world. The general problem is the assignment of too few digits to handle unique citizen identifications given normal population growth for more than 50 or 100 years.

Costs Associated With Date Repairs and Damages

The economic justifications for developing a new date and numeric information standard are the high costs which will be accrued between 1999 and 2050 for software date changes due to limitations of the current standard date and numeric formats. Let us consider the implications of the failure of the current dates.

Table 1 is a hypothetical projection of the numbers of existing software applications on a global basis which contain various kind of date and numeric problems which will require changes because of format problems.

Because dates and numeric data need to be fixed before their formats expire, but this will probably not happen, table 1 also gives an estimate of the number of applications with date problems that will miss their deadlines and not be fixed in time.

Table 1: Projected Numbers of Applications With Date Problems

Date Problem	Applications With Problem	Repaired in Time	Unrepaired Format Errors	Years of Main Impact
Year 2000	36,000,000	80.00%	7,200,000	1999-2001
Phone numbers	25,000,000	85.00%	3,750,000	2000-2025
Euro-currency	10,000,000	75.00%	2,500,000	1999-2005
Social security	15,000,000	90.00%	1,500,000	2050-2099
UNIX rollover	12,000,000	90.00%	1,200,000	2036-2038
End of file	4,000,000	90.00%	400,000	1999-2001
Leap year 2000	2,000,000	90.00%	200,000	1999-2000
GPS Rollover	250,000	98.00%	5,000	1999-2000
TOTAL	104,250,000	87.25%	16,755,000	1999-2099

Table 1 is sorted by the column labeled "unrepaired format errors" on the grounds that the greatest volume of unrepaired dates and number formats are likely to have the greatest damage potentials. Table 1 has a large margin of error but its underlying message is valid: date and numeric format problems in software are plentiful and troublesome.

As of 1998 it is obvious that it is far too late to approach 100% readiness for either the year 2000 or the Euro currency problems. At a global level, we will be lucky if even 80% of applications with date problems are ready for the year 2000. The Euro-currency situation is even stickier, and my projection here is that no more than 75% of the world's financial applications which deal with currencies will be ready in time.

Starting in 1999 and running on through 2050, computer and software date and numeric problems are going to absorb huge numbers of scarce software personnel who really should be doing more positive things. Indeed, fixing date and numeric problems is such tedious work that it hard to find people to do it without substantial pay and benefits packages.

Unfortunately a significant percentage of software personnel, possibly more than 50% of the entire software work force, will begin to spend more and more time on date and format repairs and will not be available for new applications, new functional enhancements, or work that adds positive value to business and government software.

Since the software industry has a bad track record of finishing anything on time there are strong reasons for assuming that many applications with date problems won't be updated in time.

The costs for all of these date problems will not be known with certainty until they occur, but the projections are very alarming:

- More than a trillion dollars will be spent on the year 2000 problem before it occurs, and more than two trillion dollars may be spent on damages, recovery costs, and litigation afterwards.

- More than four hundred billion dollars may be spent on Euro-currency updates, and more than six hundred billion dollars on damages, recovery costs, and litigation. Overall, the total costs associated with the Euro-currency situations could easily top a trillion dollars on a global basis if both pre-Euro costs and post-Euro damages and litigation are included.

- Given the pervasive nature of telephone communication, about a billion dollars a year is already being spent due to the frequent changes of area codes. These expenses will begin to escalate as the capacity of current telephone numbering schemes begin to approach the saturation point. The total costs can top $250 billion for software, and probably cause the premature disposal of several billion dollars worth of hand-held personal information managers (PIMs) which cannot make the transition to an expanded telephone number.

- Software cost estimates for social security numbers, the GPS date roll-over, the "nines" end of file problems, the year 2000 leap year, and for the UNIX and C library date roll-over have not been published. But all of these are likely to be in the multi-billion dollar range with the possible exception of the GPS date roll-over which may be smaller. All of these other date problems added together might top yet another trillion dollars.

The bottom line is that date and numeric format problems are becoming a black hole of software costs which will absorb far too much money and too many scarce resources. There is a strong economic justification for wanting to develop permanent date and numeric format solutions for software and computer purposes.

Summary and Conclusions

Computers and software are the major tools of business, commerce, science, and industry. Accurate date and time recording are important activities for both business and science. Also important are accurate storage of other numeric information such as telephone numbers and social security numbers.

It is obvious that computers and software are now the primary tools for numbers and for date and time recording, but unfortunately all of the current methods for handling date and time storage within computers are inadequate.

Neither the ISO nor other default date standards are fully adequate for business purposes, and are not adequate for scientific dates at all.

Neither the current GPS nor UNIX date mechanisms are fully adequate, since roll-overs due to storage limits will cause problems in 1999 and 2038 respectively.

Since computer and software date storage, date calculations, and date representation are far more important than printed dates, it would be valuable to have an international date symposium which would develop a new international standard for computerized date and time storage, and for other kinds of important numeric data such as telephone and social security numbers.

Several forms of date storage need to be included, and the standard should support both normal calendar dates and also the method of recording seconds from arbitrary starting points. This new standard should accommodate the needs of science as well as the needs of business, which means it must work over spans of billions of years in both future and past directions.

The justification for developing such a standard can be seen in the expenses that are already accumulating for the GPS roll-over, the year 2000 date problem, and will also accumulate for the UNIX 2038 problem. It would be hazardous to go forward into the 21st century without an adequate standard for dealing with dates and time storage in computers and we do not have one currently in 1998.

The enormous expenses which the software industry and all other industries are now facing due to date problems provides a strong economic reason for wanting to develop a new "super date" standard which can accommodate older date representation methods and facilitate date conversion logic among all date recording methods and all calendars.

References and Readings

Allan, David W., Ashby, Neil, and Hodge, Cliff; "Fine Tuning Time in the Space Age;" IEEE Spectrum, March 1998; pp. 42-51.

DeJager, Peter and Richard Bergeon; *Managing 00 - Surviving the Year 2000 Computing Crisis*; John Wiley & Sons, 1997.

Jones, Capers; *Assessment and Control of Software Risks*; Prentice Hall, 1994; ISBN 0-13-741406-4; 711 pages.

Jones, Capers; *Patterns of Software System Failure and Success*; International Thomson Computer Press, Boston, MA; December 1995; 250 pages; ISBN 1-850-32804-8; 292 pages.

Jones, Capers; *Applied Software Measurement*; McGraw Hill, 2nd edition 1996; ISBN 0-07-032826-9; 618 pages.

Jones, Capers; *The Year 2000 Software Problem - Quantifying the Costs and Assessing the Consequences*; Addison Wesley, Reading, MA; 1998; ISBN 0-201-30964-5; 303 pages.

Jones, Capers; *Software Quality - Analysis and Guidelines for Success*; International Thomson Computer Press, Boston, MA; ISBN 1-85032-876-6; 1997; 492 pages.

Jones, Capers; "Estimating Rules of Thumb for the Year 2000 and Euro-Currency Projects"; SPR Technical Report; Software Productivity Research, Inc.; Burlington, MA; January 1998.

Jones, Capers, "Resource Conflicts Between the Year 2000 and Euro-Currency Software Problems," Year 2000 Journal, January/February ;1998; Vol. 2 No. 1; pp. 63-70.

Jones, Keith; *Year 2000 Software Crisis Solutions*; International Thomson Computer Press, 1997.

Kappelman, Leon (editor); *Solving the Year 2000 Problem*; International Thomson Computer Press, 1997.

Kurtzman, Dr. Clifford; "Frequently Asked Questions about the Year 2000 and Similar Problems;" Tenagra Corporation, Houston, TX; December 1997; http://www.tenagra.com or http://www.year2000.com.

Lefkon, Dr. Dick (editor); *Year 2000 Best Practices for Y2K Millennium Computing: Panic in Year Zero*; Mainframe Special Interest Group (SIG) of the Association of Information Technologies (AITP); New York, NY.

Murray, Jerome T. and Murray, Marilyn M.; *The Year 2000 Computing Crisis - A Millennium Date Conversion Plan*; McGraw Hill, New York, NY; 1996.

Ragland, Bryce; *The Year 2000 Problem Solver*; McGraw Hill, New York, NY; 1997.

Robbins, Brian and Rubin, Dr. Howard; *The Year 2000 Planning Guide*; Rubin Systems, Inc.; Pound Ridge, NY; 1997.

Rubin, Dr. Howard; *Survey of Year 2000 Preparations in Fortune 500 Companies*; Meta Group, Stamford, CT; January 1998.

Ulrich, William and Ian S. Hayes; *The Year 2000 Software Crisis - Challenge of the Century*; Prentice Hall, Yourdon Press; 1997.

Yourdon, Ed and Yourdon, Jennifer; *Time Bomb 2000: What the Year 2000 Computer Crisis Means to You*; Prentice Hall PTR, Upper Saddle River, New Jersey; 1998.

YEAR 2000
To the Year 2000 Information Center™

Probabilities of Year 2000 Damages

Capers Jones

February 27, 1998

Abstract

Because the year 2000 software problem has not yet occurred, there is no absolute way of knowing what damages are going to occur or not. If global business and government software teams are energetic and successful in containing the year 2000 problem, then damages may be very slight. But if global business and government leaders are lax, then year 2000 damages can be severe.

The author is in touch with year 2000 managers and risk directors in many companies and government agencies throughout the world. This report summarizes the probabilities of various kinds of year 2000 damages occurring. The report has a high margin of error, but it is better to publish information with possible errors rather than to ignore problems which might prove to be serious.

INTRODUCTION

The year 2000 software problem has generated strongly polarized opinions in terms of the severity of the damages which might occur. On one hand are year 2000 specialists, consultants, and vendors who feel that the damages may be serious indeed. On the other hand are year 2000 researchers and some corporate officers who feel that the year 2000 problem has been exaggerated, and the damages may be comparatively minor.

Both sides agree that the year 2000 date problem affects large numbers of software applications as of 1998. They tend to disagree about two major issues:

1. How many year 2000 problems will not be fixed by the turn of the century.
1. How severe the damages will be from unrepaired year 2000 problems.

Because the year 2000 problem has not yet occurred, there is no current way of being sure exactly how many year 2000 problems will remain unrepaired at the end of the century, and what kinds of damages might result.

The situation brings to mind the uncertainty surrounding the secession of southern States after Abraham Lincoln was elected to the presidency. Many congressional and senatorial leaders thought that secession would lead to war, but there were sharply divided opinions as to how serious the conflict would be and how long it might last.

Many senators and congressmen thought that a war fought over secession might last only 30 to 90 days and would have fairly low casualties. Others feared a longer and more violent conflict. Until the war itself actually occurred, there was no way of determining which view would match the reality of the situation.

Since the year 2000 problem is still almost two years away, there is a similar polarization of opinion as to whether the remaining date problems will be serious or fairly minor. Here too, the only way of being certain is to see what happens when the 20th century ends.

Empirical Data on Software Defect Removal

For more than 50 years software defect removal operations have been less than 100% effective in finding and removing bugs. Based on studies published in two of the author's books Applied Software Measurement (McGraw- Hill 1996) and Software Quality - Analysis and Guidelines for Success (International Thomson Computer Press 1997) the average number of software errors is about five per function point. This data has been comparatively stable for the United States as a whole between the mid 1980's and 1998.

The year 2000 problem is primarily a code issue in 1998, although the problem originated in many software requirements and was also included as a design feature. However, most of the software applications which need year 2000 repairs were designed and developed years ago, so code is the primary date repository.

The U.S. average for overall defect removal efficiency, unfortunately, is currently only about 85% although top- ranked projects in leading companies such as AT&T, IBM, Motorola, Raytheon, and Hewlett Packard achieve defect removal efficiency levels well in excess of 99% on their best projects.

All software defects are not equally easy to remove. Requirements errors, design problems, and "bad fixes" tend to be the most difficult while coding errors are generally the easiest. Also, testing and inspection of source code have the largest array of tools and methodologies available. As a result, defect removal efficiency levels against coding defects are higher than for other sources of error, as can be seen by Table 1:

Table 1: United States Averages for Software Defect Removal Efficiency

(Data Expressed in Terms of Defects per Function Point)

	Defect Origins	**Defect Removal**	**Delivered**
Potentials	**Efficiency**	**Defects**	
Requirements	1.00	77%	0.23
Design	1.25	85%	0.19
Coding	1.75	95%	0.09

Document 0.60 80% 0.12

Bad Fixes 0.40 70% 0.12

Total 5.00 85% 0.75

Note that the row labeled "bad fixes" refers to secondary bugs or defects accidentally injected as a byproduct of fixing primary defects. The bad fix injection rate is seldom discussed in the year 2000 literature, but is likely to be just as troublesome as it is for other kinds of software errors.

Since the year 2000 problem is primarily a code issue and the number of year 2000 defects are smaller than the numbers of all bugs, a similar year 2000 result might resemble the following data shown in table 2:

Table 2: United States Averages for Year 2000 Defect Removal Efficiency

(Data Expressed in Terms of Defects per Function Point)

Defect Origins Defect Removal Delivered

Potentials Efficiency Defects

Year 2000 Dates 0.150 95% 0.0075

Bad Fixes 0.050 70% 0.0150

Total 0.200 95% 0.0225

Since fractional or decimal results are hard to visualize, let us scale the problem up in size and discuss the probable results of an application of 1000 function points in size, which is roughly equivalent to 105,000 COBOL statements.

For an application of 1000 function points in size, when year 2000 repairs begin there would be a total of 150 date references in this application. Assume that 95% of them would be found and repaired. The probable number of unrepaired two-digit dates still latent in the application would be roughly 8 unrepaired date fields or date calculations after the end of the century. (Note that some dates are indirectly derived and hence difficult to identify.)

However, in fixing the 142 date fields at least 10% of these repairs would contain some kind of error or "bad fix" so 14 new bugs would be injected. Of these secondary bugs, 70% would be found and repaired again but 4 of these "bad fix" bugs would slip by into the next century. The total number of unrepaired or erroneous dates at the end of the century would be approximately 12 still latent in the application. The approximate cumulative defect removal efficiency level would be 94.66% which is quite a high level for most kinds of defects, although less than desirable for the year 2000. For additional information on the topic if year 2000 defect removal, refer to the author's book The Year

2000 Software Problem - Quantifying the Results and Assessing the Consequences (Addison Wesley Longman, 1997).

Achieving High Levels of Defect Removal Efficiency

In order to achieve high levels of defect removal efficiency, it is necessary to use state of the art quality control and testing approaches. For large software applications, formal design and code inspections plus formal testing are the only known ways of exceeding 95% in cumulative defect removal efficiency levels. Only a combination of inspections and very careful testing can approach or exceed 99%, which is the level needed by the year 2000 problem.

Note that existing test libraries contain date errors too. Note also that test cases themselves are often in error. Indeed, the number of bugs or errors in test cases is often larger than the number of bugs in the applications being tested. The literature on year 2000 test errors is sparse, although this topic was cited in the author's book, The Year 2000 Software Problem - Quantifying the Costs and Assessing the Consequences (Addison Wesley Longman, 1997).

Most forms of testing are less than 50% efficient in finding errors, which is why several kinds of testing have always been needed to ensure reasonable quality levels. To ensure that defect removal efficiency levels of the year 2000 date problem approaches 99%, the following sequence of defect removal operations will be needed:

1. Formal code inspections
1. Formal test case inspections
1. Unit testing
1. Regression testing
1. Integration testing
1. Performance testing
1. System testing
1. Extended testing with clients, suppliers, etc.

If formal inspections are omitted and only a few test stages are utilized, then it is unlikely that even 90% of the year 2000 problems would be removed. Indeed, the U.S. norm for a testing sequence of unit test, regression test, integration test, and system test is usually only about 70% efficient in finding latent errors.

With possibly 5% to more than 20% of the year 2000 problems still unrepaired and remaining in software after the century ends, the probability of significant damages is alarmingly high.

Look at the numbers of latent bugs reported against every major software application: Microsoft's products, IBM products, Computer Associates, and so on. Every major software vendor realizes that less than 100% defect removal is the rule, and so they have massive support and maintenance structures in place.

It is naïve and risky to assume that 100% of year 2000 errors will be found and repaired, since the U.S. average for other kinds of bugs is only about 85% defect removal

efficiency and even "best in class" results are below 99% in efficiency.

There will be date problems in software after the century ends, and all companies and government agencies should be prepared to fix them after the end of the century when they have already done damage. This is no different from the need to provide customer support and maintenance support for software after it is deployed, because they always have bugs in them.

It would be a reasonable contingency plan to have emergency response teams available in every company and government agency to deal with the impact of undiscovered year 2000 problems.

Probabilities of Personal Year 2000 Hazards

Although the author of this report is a year 2000 consultant, he is also a homeowner, a taxpayer, a frequent flyer, and a consumer of public utilities in the forms of electricity, water, telephone services, and so forth. It is important to understand what kinds of year 2000 problems might occur and what the odds are that these can cause enough damage to be troublesome. The probabilities in table 3 are derived from the following kinds of research and analysis:

- Discussions with year 2000 repair executives
- Discussions with risk executives
- Discussions with economists
- Review of year 2000 test results in key industries such s electricity generation
- Software Productivity Research assessment and baseline studies

No matter what kind of research is performed in 1998, it is still necessary to wait until the year 2000 event itself to be absolutely certain of the outcomes. Table 3 shows the probability levels assigned by the author:

Table 3: Year 2000 Damage Probabilities Assuming Latent Date Problems

Year 2000 Problem	Probability of Occurrence
Bad credit reports due to year 2000 errors	70%
Cancellation of year 2000 liability insurance	60%
Loss of local electric power (> 1 day)	55%
Litigation against corporate officers	55%
Loss of regional electric power (> 1 day)	40%
Loss of international telephone services	35%

Errors in 2000 tax reporting (1099 forms)	35%
Errors with social security payments	35%
Errors in first January paycheck	30%
Errors or delays in tax refunds	30%
Delays or cancellations of airline flights	25%
Loss of local telephone services	20%
Errors with motor vehicle records	20%
Medical or hospital billing errors	20%
Manufacturing shut-downs (> 1 day)	20%
Process control shut-downs (> 1day)	20%
Reduction in stock values	20%
Errors in 2000 tax reporting (W2 forms)	15%
Errors in bank account balances	15%
Disruption of stock market trading	15%
Shut-down of pharmaceutical manufacturing	15%
Errors in hotel/motel reservations	12%
Delays or cancellations of shipping	10%
Errors in prescription dates	10%
Delays in UPS, FedEx deliveries	10%
Delays or cancellations of rail shipments	10%
Urban bankruptcy due to year 2000	7%
Water shortages/rationing	7%
Corporate bankruptcy due to year 2000	5%
Food shortages/rationing	3%

methods.

- Certification programs do not guarantee that 100% of year 2000 date problems will be found and removed. These certification programs merely testify that your year 2000 repair activities are prudent and energetic. There are software bugs in applications built by companies that are ISO 9001 certified and also at or above Level 3 on the Software Engineering Institute Capability Maturity Model (CMM). There will be year 2000 problems left over in companies certified by the ITAA and other year 2000 certification programs, and it would be naïve to think otherwise.

In sum, there is no reason at all to assume that year 2000 defect removal efficiency levels will be any higher than the ranges for other kinds of software errors. Software quality has been a major embarrassment to the software industry for 50 years. It is naïve to think that thousands of companies who were never very good in software quality control before the year 2000 problem will suddenly achieve higher than average defect removal rates for one of the toughest software problems in history.

Some of the major year 2000 service companies or major software producers equipped with capable year 2000 specialists, state of the art year 2000 search engines, and state of the art testing methods may be able top 99% for year 2000 defect removal efficiency levels or even hit the unlikely goal of 100% removal for well-structured applications which do not include complex or derived date logic.

However, ordinary companies attacking the year 2000 problems with untrained generalists, without year 2000 search engines, and using normal and informal testing stages are unlikely to go higher than 90% in year 2000 defect removal. If the software is particularly complex (modules > 20 in terms of cyclomatic and essential complexity) and uses indirect or derived date logic, then year 2000 repair efficiency levels can be lower than 80% and the bad-fix injection rate can top 15%.

Summary and Conclusions

The year 2000 software problem is a very difficult technical issue because software itself is a difficult topic. Since as this is written we still have almost two years to go before the year 2000 event, the exact nature of the damages are still uncertain.

If year 2000 repairs are active and energetic, then it may be possible to achieve or top 95% defect removal efficiency levels and hence face only minimal year 2000 damages.

If year 2000 repairs are sluggish and partial, which is the current situation for many organizations, then it is unlikely that more than 80% to 85% of year 2000 date problems will be found, thus leaving 15% to more than 20% of the date problems still latent at the end of the century. In this case, damages will be severe.

Readings and References

DeJager, Peter and Richard Bergeon; Managing 00 - Surviving the Year 2000 Computing Crisis; John Wiley & Sons, 1997.

Jones, Capers; Assessment and Control of Software Risks; Prentice Hall, 1994; ISBN 0-

Escheatment of bank accounts	2%
Death or injuries due to year 2000	1%

The highest probability is that we will have bad credit reports filed against us due to year 2000 problems which make us seem to be late in payments. This problem is almost certain to occur because there are just too many credit cards and billing systems in the United States for all of them to be fixed.

Another very high probability risk is that we will lose electric power for at least a day, and possibly for more than a week. Electric power plants in the United States are highly computerized and the year 2000 problem is endemic within this industry. Worse, experiments by electric companies to test out their year 2000 repairs have indicated worse problems and longer outages than anyone envisioned. These date problems are found in every form of electric generation: hydro electric, coal, and nuclear. They are often obscure and difficult to both find and fix.

Errors with government services are also in the high-risk category. The CIO of the U.S. Internal Revenue Services (IRS) has already stated that the IRS would not be ready, for the year 2000 and many other government agencies will not be ready either.

Payroll errors in January of 2000 accompanied by errors in the calculations of tax information for calendar year 1999 also have a high probability of occurring. There are too many local payroll and tax calculation software packages for all of them to be repaired.

Table 3 has a high margin of error, but several factors make at least some of the probabilities illustrated by the table distressingly likely to occur:

- In 50 years of building and maintaining software, achieving 100% rates of defect removal efficiency has never happened for large systems development or large system enhancements.
- It is highly unlikely that any company will achieve 100% efficiency in removing year 2000 problems, since many of the problems are resident in aging and poorly structured applications that are hard to test or inspect.
- It is highly unlikely that indirect dates or complex date logic can be completely isolated or removed, or even fully tested.
- The bad fix injection rates for year 2000 repairs are under-reported in the year 2000 literature, and are likely to be troublesome. Possibly as many as 10% of year 2000 repairs might create new defects. Although bad fixes are quite common, the topic of bad-fix injections is sparse in the normal quality and testing literature, and almost non-existent in the year 2000 literature.
- Errors and defects in test cases and test libraries are under-reported in the quality and test literature, and almost non-existent in the year 2000 literature other than a section in the author's year 2000 book.
- Many organizations with year 2000 software problems are not very sophisticated in software quality control and are not using state of the art defect removal

13-741406-4; 711 pages.

Jones, Capers; Patterns of Software System Failure and Success; International Thomson Computer Press, Boston, MA; December 1995; 250 pages; ISBN 1-850-32804-8; 292 pages.

Jones, Capers; Applied Software Measurement; McGraw Hill, 2nd edition 1996; ISBN 0-07-032826-9; 618 pages.

Jones, Capers; The Year 2000 Software Problem - Quantifying the Costs and Assessing the Consequences; Addison Wesley, Reading, MA; 1998; ISBN 0-201-30964-5; 303 pages.

Jones, Capers; Software Quality – Analysis and Guidelines for Success; International Thomson Computer Press, Boston, MA; ISBN 1-85032-876-6; 1997; 492 pages.

Jones, Capers; "Becoming Best in Class"; SPR Technical Report; Software Productivity Research, Burlington, MA; January 1998; 40 pages.

Jones, Capers; "Estimating Rules of Thumb for the Year 2000 and Euro-Currency Projects"; SPR Technical Report; Software Productivity Research, Inc.; Burlington, MA; January 1998.

Jones, Keith; Year 2000 Software Crisis Solutions; International Thomson Computer Press, 1997.

Kappelman, Leon (editor); Solving the Year 2000 Problem; International Thomson Computer Press, 1997.

Lefkon, Dr. Dick (editor); Year 2000 Best Practices for Y2K Millennium Computing: Panic in Year Zero; Mainframe Special Interest Group (SIG) of the Association of Information Technologies (AITP); New York, NY.

Ragland, Bryce; The Year 2000 Problem Solver; McGraw Hill, 1997.

Robbins, Brian and Rubin, Dr. Howard; The Year 2000 Planning Guide; Rubin Systems, Inc.; Pound Ridge, NY; 1997.

Rubin, Dr. Howard; Survey of Year 2000 Preparations in Fortune 500 Companies; Meta Group, Stamford, CT; January 1998.

Ulrich, William and Ian S. Hayes; The Year 2000 Software Crisis - Challenge of the Century; Prentice Hall, Yourdon Press; 1997.

Yourdon, Ed and Yourdon, Jennifer; Time Bomb 2000: What the Year 2000 Computer Crisis Means to You; Prentice Hall PTR, Upper Saddle River, New Jersey; 1998.

Contingency Planning for the Year 2000 Software Problem

Capers Jones

December 5, 1997

Abstract

It is becoming obvious that many software applications with date problems will not be repaired before the end of the century. It is even more likely that data bases with two-digit date fields will not be repaired in time. Thus the year 2000 problem is likely to have serious consequences for many citizens, businesses, and government organizations. This means that it is time to begin contingency planning for dealing with the kinds of problems which are likely to occur when computers and software applications begin to fail or produce incorrect results.

The most serious problems in terms of daily life can be anticipated with software failures which affect our national "infrastructure" or basic services such as telephone switching, electric power, air travel, and transportation of food and goods. However, year 2000 problem can also affect financial calculations, tax returns, annuities, and other forms of business software.

Due to the lack of visible and effective leadership by government organizations, contingency planning for dealing with the year 2000 problem needs to be addressed by businesses and individual citizens.

INTRODUCTION

Surveys of software executives and financial officers in both the United States and in Europe return similar and unfortunate results: A significant number of software applications with date problems will not be repaired before the end of the century. Because repairs to date fields in data bases are even more difficult than software repairs, it is very unlikely that data bases will be fully year 2000 compliant by the end of the century.

Given the strong probability that somewhere between 10% and perhaps 35% of potential year 2000 software problems will still be present at the dawn of the next century, it is now time to begin to start contingency planning for minimizing the damages which unrepaired year 2000 problems might cause.

There are many kinds of year 2000 problems which individual companies cannot deal with. For example, if the year 2000 problem causes an extended loss of electric power combined with a loss of telephone services, then business as we know it will basically stop until these essential services can be restored.

Because year 2000 compliance will be less than 100%, it is quite probable that the costs in the next century for damages, recovery, and litigation will far exceed the costs of achieving the partial levels of year 2000 compliance which I estimate to be about 85% in the United States but 75% or less elsewhere. The following table shows predicted year 2000 costs at a global level for expenses both before and after the year 2000 event:

Global Total of Pre-2000 and Post-2000 Repairs and Recovery

Global function points deployed	10,212,750,000
Global LOC deployed	1,164,253,500,000

Year 2000 Topic	Global Year 2000 Cost Elements	Cost per Func. Pt.	Cost per LOC
Expenses Prior to 2000 AD			
Initial software repairs	$530,000,000,000	$51.90	$0.46
Secondary "bad fix" software repairs	$50,000,000,000	$4.90	$0.04
Test library repairs	$75,000,000,000	$7.34	$0.06
Data base repairs	$454,000,000,000	$44.45	$0.39
Hardware chip replacements	$76,000,000,000	$7.44	$0.07
Hardware performance upgrades	$150,000,000,000	$14.69	$0.13
Subtotal	$1,335,000,000,000	$130.72	$1.15
Expenses After 2000 AD			
Litigation and damages	$300,000,000,000	$29.38	$0.26
Post-2000 damages	$580,378,125,000	$56.83	$0.50
Post-2000 recovery expenses	$1,405,843,750,000	$137.66	$1.21
Subtotal	$2,286,221,875,000	$223.86	$1.96

TOTAL	$3,621,221,875,000	$354.58	$3.11

If this rather uncomfortable prediction is true, then the expenses after the year 2000 problem will be almost twice those expended before the year 2000 problem occurs and begins to wreak havoc with global economies.

It is interesting that although my projected total costs are higher than the predictions of other organizations such as Gartner Group and Giga Group, almost every one of my line-item costs are lower. For example, my software repair cost estimate is lower than Gartner Group while my litigation cost estimate is lower than Giga Group. My data has a higher total because I include cost elements such as hardware upgrades and data base repairs which are not part of the other year 2000 cost predictions.

The following contingency planning discussions are excerpts from my book, The Year 2000 Software Problem: Quantifying the Costs and Assessing the Consequences (Addison Wesley Longman 1998).

PREPARATIONS DURING 1999

Assuming that your company has a year 2000 project office in place, this office will be a major component for year 2000 damage control. Indeed, many year 2000 project offices will be larger in the first half of the year 2000 than they were in 1999!

One of the critical tasks of your year 2000 project office will be to assemble a master list of the year 2000 contact points of every client, supplier, bank, and government agency with which your computers share data. The year 2000 problem is a global issue and it will impact all organizations simultaneously. Therefore it is critical to have contact information available before the end of calendar year 1999.

Because of possible loss of telephone and electricity due to the year 2000 problem, your contact list should include all forms of contact with other year 2000 project offices; i.e. regular phone numbers, cellular phone numbers, pager numbers, fax numbers, email addresses, web site URLs, and so forth.

Although you will probably not publish the information or make it widely known, your year 2000 project office should also have information on the home telephone numbers and private email addresses of your own executives, and of the year 2000 managers and technical staffs within both your organization and key suppliers.

Year 2000 Status Reporting

Your year 2000 project office should also publish frequent status reports during 1999 on both your own status, and also on the preparedness levels of the organizations with whom you do business. In the last quarter of 1999 your year 2000 project office should assemble and produce weekly updates of a year 2000 status report that includes:

- Your own year 2000 readiness levels by business unit.
- The year 2000 status of your telephone company.
- The year 2000 status of your electric power company.
- The year 2000 status of your banks.
- The year 2000 status of airlines and air traffic control.
- The year 2000 status of shipping services.
- The year 2000 status of rail and surface travel.
- The year 2000 status of the postal service.
- The year 2000 status of software vendors whose products you utilize.
- The year 2000 status of data bases, data warehouses, and repositories you use.
- The year 2000 status of government agencies to whom you supply data.
- The year 2000 status of government agencies from whom you receive data.
- The year 2000 status of your major suppliers.
- The year 2000 status of your major clients.
- The year 2000 status of your major competitors.

In a perfect world, quite a lot of status information would be published by government agencies. Unfortunately government agencies are not among the leaders in year 2000 readiness, and indeed are often found among the laggards. Do not expect much help from government agencies, and be prepared for major year 2000 failures among the key government agencies with whom you do business.

Every one of your employees will be anxious about possible year 2000 problems and so your year 2000 project office should be a focal point for the latest and most accurate information as the century comes to a close.

Manual Back-Ups For Automated Processes

Because a significant number of automated processes may be at least temporarily damaged or suspended when the year 2000 problem occurs, all organizations should prepare manual back-up procedures for key business activities, such as:

- Banks may need extra tellers and extra accountants to handle manual transactions, assuming that automatic teller machines may have problems, or other computerized systems may be affected.

- Since the year 2000 problem can put an abrupt end to many data base transactions and affect data bases, data warehouses, OLAP, data mining, and any other form of long-range data analysis you need to prepare to exist without these capabilities for a period that can run from a few weeks to perhaps a year in the worst case scenario.

-
- Retail stores and restaurants need a contingency plan in case credit-card and check validations are suspended. (A "cash only" policy may limit damages, but will offend clients.)

- Retail stores and restaurants should plan to have calculators available for cashiers,

and should instruct them in the basics of totaling bills without using a computer.

- Police departments should prepare for extra shifts and extra duty, assuming that things such as traffic light controls may temporarily shut down or power outages may disrupt all forms of traffic control.

- Fire departments should prepare for extra shifts and extra duty, assuming that the year 2000 problem triggers a number of false alarms from computerized fire detection systems.

- Electric power and telephone companies should prepare for 24-hour emergency repairs at the end of the century. (Hopefully the year 2000 problem will not occur at the same time as a major blizzard, but December 31 is a time of frequent heavy snow.)

- Local governments should prepare to stay open all night on December 31, 1999 and January 1, 2000 because of the possibilities of unexpected problems with urban services.

No doubt in the last few months of 1999, every newspaper, television station, radio station, and Web site in the world will begin a year 2000 count-down, with additional information, advice, and warnings.

A ONE-YEAR CONTINGENCY CALENDAR FOR 2000

The following planning calendar highlights the major contingency activities for the critical period running from January 1, 2000 through December 31 of the same year. The calendar is aimed at businesses which use computers for production, finance, sales, marketing, and personnel applications.

The calendar assumes that your enterprise has a formal year 2000 damage control team identified, and that they will be fully staffed and ready to go by the last day of 1999.

January 2000: Immediate Damage Control

- All critical software systems monitored for year 2000 compliance.
- All critical data base queries monitored for year 2000 compliance.
- All suppliers and contractors monitored for year 2000 compliance.
- Physical problems corrected (elevators, security systems, phones, faxes, etc.)
- Manual back-up methods deployed if needed (check writing, invoices, etc.)
- Emergency year 2000 repairs for undetected problems.
- Year 2000 client-reported problems corrected.
- Year 2000 subcontractor problems reported and corrected.
- Year 2000 repair vendors notified of problems they missed.
- Analysis of legal and liability status (your enterprise's own hazards.)
- Analysis of legal and liability status (damages to your enterprise)
- Analysis of possible tax errors performed.

- Analysis of billing and accounting errors performed.
- Post-2000 repair cost estimate updated.
- Post-2000 insurance claims analyzed.
- Directors and officers liability status analyzed.

February Through July 2000: Damage Assessment and Litigation Filing

- Initial claims made for damages under year 2000 insurance.
- Possible filing of lawsuits by clients for year 2000 problems.
- Possible filing of lawsuits by shareholders for year 2000 problems.
- Possible filing of lawsuits by employees for year 2000 problems.
- Possible filing of lawsuit by tax authorities for year 2000 problems.
- Possible filing of lawsuits by your company for botched year 2000 repairs.
- Analysis of overall liabilities for enterprise, directors, and officers.
- Performance monitoring and performance tuning of applications.
- Temporary repairs (i.e. encapsulation) replaced by date field expansion.
- Manual back-up methods phased out if possible.

August Through December 2000: Recovery and Replacement

- Litigation disclosures and depositions begin.
- Some companies may begin bankruptcy filings.
- Hardware upgrades deployed to restore lost performance.
- Continued replacement of "masking" year 2000 repairs with date field expansions.
- Overall year 2000 costs accumulated, with estimates for remaining costs.
- Overall year 2000 costs disclosed in financial reports to shareholders.
- Some year 2000 repair teams return to other work.
- Some suspended software projects resume.

Calendar year 2000 is likely to have an unpleasant beginning as the year 2000 date problems erupt throughout the global business and government community. Hopefully the problems can be controlled and eliminated without major disruption of key services and local infrastructures.

The human race is never very effective in disaster prevention, but often demonstrates remarkable resiliency and vigor in damage control and recovery. The year 2000 problem is following a fairly typical pattern, in that preventive activities are probably insufficient to head off significant problems. Hopefully governments and businesses will not be shut down so severely that recovery borders on being impossible. After the year 2000 problems do occur, then we are likely to witness full scale mobilization by government agencies and hopefully a fairly rapid recovery period.

However, the year 2000 repair and damage expenses are likely to be high enough so that many companies will report financial losses during calendar years 1998, 1999, and 2000. This, in turn, can depress share values, reduce tax revenues, and have other serious implications too that are outside the scope of this article on contingency planning.

Another topic outside the scope of this article is how to deal with the anticipated deluge of lawsuits that will occur in the wake of the year 2000 problem. However, every

corporate executive should be seeking legal advice right now, because many of them will face personal liability issues if year 2000 problems lower stock values or cause damage to clients.

PERSONAL YEAR 2000 CONTINGENCY STRATEGIES

Perhaps the best way to deal with personal contingency planning for the year 2000 problem are to start with some of the defensive actions which the author is considering.

- Keep paper back-up records of bank balances, credit card balances, investments, and utility bills for telephones, water, electricity, gasoline, cable television, car payments, mortgage payments, internet service, etc.

- If you receive any strange communications from companies or government agencies where a date of "00" is cited, contact the issuing office and try and get the error corrected. For example, you might get a bill or invoice which asserts that you are 99 years late for a payment of some kind.

- Check with your personal bankers and brokers to ascertain their year 2000 status. There is no value in checking before early 1999 because the final status will not be known until then. The more proactive banks and financial institutions will not only know their status, but will probably have major ads and even billboards saying things like, "Acme National Bank is Year 2000 Compliant" in order to assure customers that they have things under control.

- If you invest in stocks, bonds, commodities, futures, derivatives or other intangibles you should evaluate the year 2000 compliance of your brokers and the safety of your investments starting at once. The impact of the year 2000 on stocks, bonds, etc. is uncertain as this article is written. Companies that achieve year 2000 compliance may even see an increase in stock value, but problems of the magnitude of the year 2000 event will have unpredictable consequences.

- Look for announcements by your local telephone and electric companies or contact their consumer liaison offices to find out if they will be year 2000 compliant.

- If you are taking prescription medicines, be sure to have your prescriptions refilled before the end of the century. Check with your pharmacy about year 2000 compliance and when your next prescription refill dates will come up.

- If your home uses oil heat and you depend upon computer-controlled automatic deliveries by your oil company, check with your oil company to be sure that their delivery schedule software packages are year 2000 compliant. The month of January 2000 is not a good time to run out of heating oil.

- Some of the year 2000 problems with electrical generating equipment may cause voltage fluctuations and even changes in electric frequency from 60 hertz to something else. While surge protectors might help, they also might not be

sufficient. Check announcements from your power company about year 2000 compliance, and if they don't know, you should perhaps unplug electric equipment that that might be running at midnight on December 31, 1999 such as computers, stereo equipment, refrigerators, and other items.

- Check with company controllers and finance groups to ascertain that your income tax information will be handled correctly for 1999 and 2000.

- If you discover in January of 2000 that your income tax withholding (W2 forms in the U.S.) or interest calculations (1099 forms in the U.S.) are in error, contact both the originating offices to report the error, and also a tax attorney or certified tax preparation specialist to ensure that the errors will not affect your personal tax status. You might also call the Internal Revenue Service (IRS) or equivalent organization since they will probably have a year 2000 contact point available.

- In the first quarter of 2000 AD, check your credit rating to ensure that no adverse credit information has been filed against you due to year 2000 problems. It would be prudent to check in the last quarter of 1999 and keep a copy of that report also.

- If your company has badge-lock security systems on the doors, check with your security group to be sure that the software controlling your entry and exit points is year 2000 compliant, and will also work on February 29, 2000.

- If you work in a large office building with elevators, check with your building maintenance or security office to see how they plan to handle elevator inspections. Some elevators keep computerized records of inspection dates and will shut down if the inspection date has passed. Moving from 1999 to 2000 will probably cause an indication that the inspection period has past, thus causing the elevators to descend to the first floor and shut down.

- If your office computer systems and networks use security systems, check with your system administrator to find out if your security packages are year 2000 compliant and won't lock you out on either January 1, 2000 or on February 29, 2000.

- Keep some emergency funds available in the form of travelers checks or cash, or both, in case bank ATM's or credit-card processing are affected for the first week of January of 2000.

- Attend town meetings and raise the question of year 2000 compliance with local governments. In particular, inquire how much the year 2000 repairs will costs, and from what source the funds or budget have been provided.

- If you need to renew your driver's license, your automobile registration, or have any other business with state and local governments in December of 1999 or January of 2000, try and get it done as early as possible, say the first week of December, since the offices will perhaps be swamped with calls and requests for year 2000 information as the end of the year approaches.

- Check for FAA, airline, and rail travel announcements about year 2000 compliance in the last few months of 1999.

- If you plan to travel over the New Year's holiday, check with travel agents and airlines to be sure that schedules for arrivals and departures will successfully handle year 2000 and leap-year situations. This is especially true for European travel, since Western Europe is lagging in year 2000 repairs.

- If you plan to travel by automobile, fill up your gas tank on or before December 31, 1999 in case electric power shuts down, or gasoline credit-card reading instruments become inoperative.

- Check the year 2000 announcements on local television and radio, and also in local newspapers. As the end of the century approaches, these channels of information will devote increasing amounts of air time and column space, and have much better information available, than is currently the case.

These are merely precautionary measures, and do not actually indicate that the problems will occur. Hopefully all major banks, utilities, airlines, and local governments will achieve such a level of year 2000 compliance that the precautions will prove to be unnecessary. However, it is better to take early precautions and not need them rather than discovering after January 1, 2000 that you should have been more careful.

SUMMARY AND CONCLUSIONS

The year 2000 problem is one of the largest business problems ever to face the human race. The magnitude of the year 2000 problem approaches some natural disasters such as earthquakes, hurricanes, and floods. In general, humans are inadequate in terms of defect prevention, but often energetic and effective in disaster recovery activities.

Government agencies are microcosms of the human species and often magnify both our strengths and weaknesses. In the context of the year 2000 problem, the actions of government agencies at all levels runs from utter incompetence through fairly sophisticated preparation and planning, but on average governments have not been very effective in dealing with the year 2000 event.

Since it is becoming increasingly obvious that perfect year 2000 compliance will not occur, it is now time to address contingency plans or how to deal with the year 2000 problems that are discovered in January of 2000 AD.

Although it is apparent we cannot stop all year 2000 problems from occurring, it is hopeful that we can recover as quickly in their aftermath as the human race often does in the wake of natural disasters such as floods and earthquakes.

REFERENCES AND READINGS

Testimony of Ann Coffou, Managing Director of Giga Information Group before the U.S. House of Representatives Science Committee (March 20, 1997); Giga Information Group; One Kendall Square, Cambridge, MA

DeJager, Peter and Richard Bergeon; Managing 00 - Surviving the Year 2000 Computing Crisis; John Wiley & Sons, 1997.

Emery, Martyn; *Predicting the Impact of Hurricane 2000*; Insurance Specialist, Vol. 2, No. 6; 1997; page 60.

Guenier, Robin; *Reports of the demise of Taskforce 2000 may be premature*; Insurance Specialist; Vol. 2, No. 6; 1997; page 55.

Hall, B. and Schick, K: *The Year 2000 Crisis - Estimating the Cost*; Gartner Group, Stamford, CT.; Application Development and Management Group (ADM) Research Note; Key Issue Analysis; KA 21-1262; 1997; (revised from time to time).

Jinnett, Jeff; *Legal Issues Concerning the Year 2000 "Millennium Bug*;" LeBoeuf, Lamb, Greene & MacRae L.L.P., New York, NY; published in The Computer Lawyer, Vol. 16, 20-200; December 1996.

Jones, Capers; *Geriatric Care for Aging Software*; SPR Knowledge Base, Vol. 1 No. 1, 1994; Software Productivity Research, Burlington, MA.

Jones Capers; *Software Enhancement Modeling*; Software Maintenance: Research and Practice, Vol. 1. 91-100; 1989.

Jones, Capers; *Finding Time for Year 2000 Repairs*; Special Insert edition of Software Development; Vol. 6, No. 1, January 1998; pp. S1-S8.

Jones, Capers; The Year 2000 Software Problem - Quantifying the Costs and Assessing the Consequences; Addison Wesley Longman, Reading, MA; 1998; 315 pages.

Kendall, Robert C: *Year 2000: How to Make the Problem Smaller, How to Make it Bigger;* American Programmer magazine; Vol. 10, No. 6; June 1997; pp. 25-28.

Kappelman, Leon (Editor); The Year 2000 Problem - Strategies and Solutions From the Fortune 100; International Thomson Computer Press, Boston, MA; 1997; 447 pages.

Keyes, Tony; The Year 2000 Computer Crisis - An Investor's Survival Guide; The Y2K Investor, Brookeville, MD; 1997; 314 pages.

Levy, Steven and Haffner, Katie; *The Day the World Crashes*; Newsweek; June 2, 1997.

Parsley, David and David Smith; *White House is Warned of EMU Timebomb*; The Sunday Times (London); November 23, 1997.

Paul, Joan; *International Tax Consequences of Year 2000 Fix Costs*; June 1997 article of the month; de Jager Year 2000 web site; http;//www.year2000.com/archive/taxes.html.

Peraino, Vito; *Corporate Director Liabilities and the Year 2000 Problem*; Delaware Corporate Litigation Reporter, February 27, 1997; Also a technical report by the law company of Hancock, Rothert & Bunshoft, Los Angeles.

Redman, Judy (Editor); *Computer Preparations for Euro Trail Behind Year 2000*; Viasoft News Release; Viasoft Inc., Phoenix AZ; November 24, 1997.

Reid, Warren; *The Year 2000 Crisis*; American Programmer, Vol 10. No. 6; June 1997; pp. 31-41.

Robbins, Brian and Rubin, Dr. Howard; The Year 2000 Planning Guide; Rubin Systems, Inc.; Pound Ridge, NY; 1997.

Ulrich, William and Ian S. Hayes; The Year 2000 Software Crisis - Challenge of the Century; Prentice Hall, Yourdon Press; 1997.

Go to:
[Papers][Status Reports][Biographies][Questions/Comments][Hot Links]
[International Directory][Conference Home Page]
[Y2KHome][CIO Info Directory]
[Back to Top]

AVERY

BUSINESS PLANS

AKSARBEN Chapter AITP Presents

infotec '99

Information Technology Conference & Exhibition

Home
Overview
Registration
Lodging & Airfare
Calendar
Keynotes
Seminars
Exhibitors
Emporium
User Groups
Museum
Banquet
The Management
Omaha

A TIME LINE OF COMPUTER, BUSINESS AND ASSOCIATION HISTORY

3000 BC: Dust abacus is invented, probably in Babylonia.

1800 BC: Babylonian mathematician develops algorithms to resolve numerical problems.

1000 BC: Phoenicians encode language into symbols - the phonetic alphabet is developed.

Piles of stones are used to Represent 'counting' of objects-an abstract concept.

'Abaci' tables are developed using positional notation.

Paper is invented.

500 BC: Bead and wire abacus is created.

'Abacus' first personal calculator -introduces the idea of letting one object equate to several others –Babylonia.

200 AD: Saun-pan computing tray is used in China; soroban computing tray used in Japan.

700-800. - Arabic numerals spread through Europe introduces concept of the 'zero' & 'fixed places' (Roman numerals remain in some places until 17 century)

1000: Gerbert of Aurillac or Pope Sylvester II devises a more efficient abacus.

1100 - Mukhammed ibn Musa Al'Khowarizmi develops concept of following a written process to achieve a goal - the 'algorithm' is born

1400 - Korea begins using mobile characters for printing

1457 Gutenberg Printing Press is invented.

1612: John Naiper, Baron of Merchiston, Scotland - first printed use of the 'decimal point' and develops the 'bones' system of multiplication. 1614 - defines log tables.

1622: William Oughtred develops the slide rule in England.

1623: William Schickard, prof. Tubingen, Germany - builds first 'adding machine' based on Naiper's studies

1624: Wilhelm Schickard builds first four-function calculator-clock at the University of Heidelberg.

1642: Blaise Pascal builds the first numerical calculating machine in Paris.

1673: Gottfried Leibniz builds a mechanical calculating machine that multiplies, divides, adds and subtracts. It is a stepped cylindrical gear to do multiplication by successive additions to an accumulator.

1780: American Benjamin Franklin discovers electricity.

1801: Joseph-Marie Jacquard invents perforated card for use on his loom. People riot against machine technology.

1822: In England Charles Babbage designs a Difference Engine to calculate logarithms for navigation tables, but the machine is never built.

1833: Charles Babbage designs the Analytical Machine that follows instructions from punched-cards. It is the first general purpose computer.

1842: Lady Ada Byron, Countess of Lovelace and daughter of Lord Byron, the poet, documents Babbage's work and writes programs for Babbage.

1844 - Samuel Morse sends message from Washington to Baltimore - 36 miles - by wire. "What hath God wrought?"

1854: Irishman George Boole publishes The Mathematical Analysis of Logic using the binary system now known as Boolean algebra.

1855: George and Edvard Scheutz of Stockholm build the first practical mechanical computer based on Babbages work.

1876: Telephone is invented by Alexander Graham Bell.

1884: Herman Hollerith applies for patents for automatic punch-card tabulating machine.

1884: Institute of Electrical Engineers (IEE) is founded.

1886: William Burroughs develops the first commercially successful mechanical adding machine.

1889: Patent is issued for Hollerith tabulating machine.

1890 - Herman Hollerith - in response to U S Census Bureau submits (and wins) the bid for automating census compilation using punched card techniques.

1903: Nikola Tesla, a Yugoslavian who worked for Thomas Edison, patents electrical logic circuits called gates or switches.

1911: Computer-Tabulating-Recording Company is formed through a merger of the Tabulating Company (founded by Hollerith), the Computing Scale Company, and the International Time Recording Company, later followed by merger into International Business Machines.

1912: Institute of Radio Engineers (IRE) is formed.

1914: Thomas J. Watson becomes President of Computing-Tabulating-Recording Company.

1921: Czech word robot is used to describe mechanical workers in the play R.U.R. by Karel Capek.

1924: Computing-Tabulating-Recording Company changes its name to International Business Machines.

By 1925 much of the early work in computing theory had been lost, or was never sufficiently recorded, therefore considerable computational knowledge was re-discovered during the next two decades.

1925: Vannevar Bush, builds a large scale analog calculator, the differential analyzer, at MIT.

1927: First public demonstration of television. Radio-telephone becomes operational between London and New York.

1927: Powers Accounting Machine Company becomes the Tabulating Machines Division of Remington-Rand Corp.

1928: A Russian immigrant, Vladimir Zworykin, invents the cathode

ray tube (CRT).

1931: First calculator, the Z1, is built in Germany by Konrad Zuse.

1933: First electronic talking machine, the Voder, is built by Dudley, who follows in 1939 with the Vocoder (Voice coder).

1936 - Konrad Zuse, Berlin, Germany, Zuse-1 begin developing a relay calculator using binary arithmetic.

1936: Englishman Alan M. Turing while at Princeton University formalizes the notion of calculable ness and adapts the notion of algorithm to the computation of functions. Turing's machine is defined to be capable of computing any calculable function.

1937: George Stibitz builds the first binary calculator at Bell Telephone Laboratories.

1938: Hewlett-Packard Co. is founded to make electronic equipment.

1938 - Helmut Schreyer and Zuse, perform the first Z-1 calculation and begin Z-2.

1939 - Stibitz develops a large scale electro-mechanical Complex Number Calculator for Bell Labs. A year later the Bell Labs Model I is the first computing machine connected remotely via telephone lines to another device. World War II is the impetus for much advancement in automatic calculation and computing.

The need for code encryption/decryption, ballistics & firing calculations and navigation tables drive the efforts.

1939: First Radio Shack catalog is published.

1939: John J. Atanasoff designs a prototype for the ABC (Atanasoff-Berry Computer) with the help of graduate student Clifford Berry at Iowa State College. In 1973 a judge ruled it the first automatic digital computer.

1940: At Bell Labs, George Stibitz demonstrates the Complex Number Calculator, which may be the first digital computer.

1940: First color TV broadcast.

1940: Remote processing experiments, conducted by Bell Laboratories, create the first terminal.

1941: Atanasoff visits IBM only to hear that "IBM sees no future in

electronic computing.

1941: Konrad Zuse builds the Z3 computer in Germany, the first calculating machine with automatic control of its operations.

April 9, 1943 proposal paper - John William Mauchly and John Presper Eckert, under guidance from John Brainerd, Dean of the Moore School of Electrical Engineering, University of Pennsylvania, begin development of the Electronic Numerical Integrator And Calculator - ENIAC computing machine on behalf of the US Army, Ballistic Research Laboratory.

John von Neuman visits Mauchly & Eckert and later develops paper on their work.

1944: Team at Bletchley Park, England, builds a decryption machine, Colossus, based on the U.S. ENIGMA machine used earlier. Colossus is used in planning for D-Day and plays critical role in Allies success.

Team includes Alan Turing. and M.H.A. Neuman

Existence of Colossus is kept secret until 1970.

Decryption algorithms are kept secret even longer.

1944 Harvard University - Mark I - first large scale general purpose electro- mechanical calculator. Conceived by Howard Aiken and implemented by IBM researchers. The machine, sponsored by the Navy, is also known as the IBM Automatic Sequence Control Calculator (ASCC).

Program is not internally stored but driven by paper tape.

1944 Grace Murray Hopper, later known as the 'First Lady of Computing', joins Aiken at Harvard. She is the third programmer assigned to the Mark I.

1944: Colossus Mark II is built in England.

1944: Mark I (IBM ASCC) is completed, based on the work of Professor Howard H. Aiken at Harvard and IBM. It is a relay-based computer.

1945: John von Neumann paper describes stored-program concept for EDVAC.

1945 September 9, 1945, 3:45 P.M. - Grace Hopper, working in temporary, *windows-open*, W W I building at Harvard University, finds and removes a 'computing problem' from the relay switches of

the Mark II. It is a large moth smashed in Relay # 70 on Panel F.
From that point forward, fixing compute problems becomes known as 'debugging.'

Arthur C. Clarke publishes his work *"Extra-Terrestrial Relays"* describing the use of geostationary satellites to provide worldwide communications. The telecommunications satellite is conceived.

1946: Binac (Binary Automatic Computer), the first computer to operate in real time, is started by Eckert and Mauchly; it is completed in 1949.

1946: ENIAC (Electronic Numerical Integrator and Computer), with 18,000 vacuum tubes, is dedicated at the University of Pennsylvania. It was 8 by 100 feet and weighed 80 tons. It could do 5,000 additions and 360 multiplications per second.

1946: Eckert-Mauchly Computer Corporation is formed as the Electronic Control Co. to design a Universal Automatic Computer (Univac).

1946: Term bit for binary digit is used for first time by John Tukey.

1947: Alan M. Turing publishes an article on Intelligent Machinery which launches artificial intelligence.

1947: Association for Computing Machinery (ACM) is formed.

1948: EDSAC (Electronic Delay Storage Automatic Calculator) is developed at the University of Cambridge by Maurice V. Wilkes.

1948: IBM introduces the 604 electronic calculator.

1948: IBM builds the Selective Sequence Electronic Calculator (SSEC), a computer with 12,000 tubes.

1948: Transistor is invented by William Bradford Shockley with John Bardeen and Walter H. Brattain.

1949: EDVAC (Electronic Discrete Variable Automatic Computer) supports the first tests of magnetic disks.

1949 An Wang, Harvard, patents the concept of core memory (single wire, delay line technology)

1949: Jay Forrester uses iron cores as main memory in Whirlwind. Forrester patent is issued in 1956.

1949: Claude Shannon of MIT builds the first chess playing machine.

1949: Hiroshi Yamauchi takes over as president of Japanese domestic playing card company. The company's name is Nintendo.

1949 : First UNIVAC computer is delivered to the US Census Bureau. Initially over budget and late, 46 more are eventually built. Stores 12,000 digits in random access mercury-delay lines.

1950: Remington-Rand acquires Eckert-Mauchly Computer Corp.

1950: SEAC (Standards Eastern Automatic Computer) is delivered to the National Bureau of Standards.

1951: December 26, The National Machine Accountants Association (NMAA) was founded and chartered in Chicago, Illinois. This group was the precursor to DPMA.

1951: First Joint Computer Conference is held.

1951 Howard Aiken's Mark II is delivered to the Naval service Weapons Center. First full scale machine to feature drum memory. Mark II is the first computer pictured on the cover a magazine (TIME).

1951 Coronado Corporation changes its name to Texas Instruments, Inc.

1951: Maurice Wilkes, Stanley Gill and David Weaver realize the difficulties of programming a computer and develop the concept of subroutines as well as the first textbook on programming computers.

1951: IEEE Computer Society is formed.

1951: UNIVAC I is installed at the Bureau of Census using a magnetic tape unit as a buffer memory.

1951: Wang Laboratories, Inc. is founded by An Wang in Boston.

1951: Whirlwind computer becomes operational at MIT. It was the first real-time computer and was designed by Jay Forrester and Ken Olsen.

1952: The first annual NMAA convention was held in Minneapolis.

1952 Grace Hopper, presents a paper on "The Education of the Computer" and describes the concept of compilers and the language translators.

1952: Fred Gruenberger writes first computer manual.

1952: IBM introduces the 701, its first electronic stored-program computer.

1952: Nixdorf Computer is founded in Germany.

1952: Remington-Rand acquires Engineering Research Associates (ERA).

1952: RCA develops Bizmac with iron-core memory and a magnetic drum supporting the first database.

1952: A fake UNIVAC front panel is used for the televised CBS election coverage. Actual connection is to Remington-Rand in Phil, PA. The UNIVAC predicts the outcome with 5% of the vote in just one hour after the polls close. An Eisenhower landslide

1952 G.W. Dummer, British radar expert, proposes electronic equipment be manufactured as a solid block - no interconnecting wires. Prototype fails and he receives little support for research.

1952: U.S. Department of Justice sues IBM for monopolizing the punched-card accounting machine industry.

1953: AK-SAR-BEN Chapter of the National Machine Accountants Association (NMAA) was formed. E. Stuart Johnson - President.

1953: Burroughs Corp. installs the Universal Digital Electronic Computer (UDEC) at Wayne State University.

1953: First high-speed printer is developed by Remington-Rand for use on the Univac.

1953: First magnetic tape device, the IBM 726, is introduced with 100 character-per-inch density and 75 inches-per-second speed.

1953: IBM ships its first stored-program computer, the 701 for United Nations in Korea. It is a vacuum tube, or first generation, computer.

1954: FORTRAN is created by John Backus at IBM. Harlan Herrick runs the first successful FORTRAN program.

1954: Gene Amdahl develops the first operating system, used on IBM 704.

1954 Texas Instruments announces start of commercial production on silicon transistors.

1954 Commodore is founded by Jack Traimel as a "typewriter repair

service

1955: The Lincoln -CORNHUSKER Chapter of NMAA, the National Machine Accountants Association was formed. J. Max Hoffmann - Pres. 14 persons / companies attended the first meeting.

1955 The first transistor calculator, TRADIC, is built in the Bell Telephone Laboratories by J.H.Felker

1955: First SHARE users group meeting is held. AUG22

1955: Remington-Rand merges with Sperry Gyroscope to form Sperry-Rand.

1955 IBM 704 introduced. First commercial machine w/ floating point hardware. Gene Amdahl is chief architect. Also in 1955

February 24, 1955 - Mrs. Jobs names her baby boy - "Steven"

October 28, 1955 - Mrs. Gates names her baby boy - "William"

1956: APT (Automatic Programmed Tool) is developed by D.T. Ross.

1956: Burroughs acquires Electrodata and the Datatron computer, which becomes the Burroughs 205.

1956: Government antitrust suit against IBM is settled; consent decree requires IBM to sell as well as lease machines.

1956: A. Newell, D. Shaw and F. Simon invent IPL (Information Processing Language.)

1956: RCA ships the Bizmac.

1956: T.J. Watson, Jr. assumes presidency of IBM.

1956: The acronym artificial intelligence is coined by John McCarthy.

1957: Control Data Corporation is formed by William C. Norris and a group of engineers from Sperry-Rand.

During the latter half of the 50's vacuum tube technology gave way to the transistor and the 'first era of computers' came to an end.

1957: DEC, Digital Equipment Corporation is founded by Ken Olsen.

1957

USSR launches Sputnuik. In response, U.S. forms the Advanced

Research Projects Agency (ARPA) to establish U.S. lead in military science & technology.

1957: First issue of Datamation is released.

1957: Honeywell joins with Raytheon to ship the Datamatic 1000.

1958: ALGOL, first called IAL (International Algebraic Language), is presented in Zurich.

1958: First virtual memory machine, Atlas, is installed in England by Feranti. It was developed at the University of Manchester by R.M. Kilburn.

1958: First electronic computers are built in Japan by NEC: the NEC-1101 and -1102.

1958: Frank Rosenblatt builds the Perceptron Mark I using a CRT as an output device.

1958: LISP is developed on the IBM 704 at MIT under John McCarthy.

1958: Seymour Cray builds the first fully transistorized supercomputer for Control Data Corp., the CDC 1604.

1958 - Jack St. Claire Kilby (Texas Instruments) conceives and proves idea of integrating transistors with resistors and capacitors on a single semi-conductor chip.

1958 - Whirlwind becomes reality as SAGE System for Air Defense

1959: COBOL is defined by the Conference on Data System Languages (Codasyl), based on Grace Hoppers Flow-Matic.

1959: First packaged program is sold by Computer Science Corporation.

1959: IBM introduces the 1401. Over 10,000 units will be delivered during its lifetime.

1959: IBM ships its first transistorized, or second generation, computers, the 1620 and 1790.

1959: General Electric develops machine to recognize Magnetic Ink Code Recognition (MICR) for Bank of America. Its a high water mark for GE computing.

1960: The NMAA sponsored a special meeting and established the

CERTIFICATE in DATA PROCESSING, a professional examination program.

1960: Benjamin Curley develops and ships the first minicomputer, the PDP-1, at Digital Equipment Corporation.

1960 - Grace Hopper, Joe Wegstein & an industry committee develop the Common Business Oriented Language -COBOL. (ALGOL 60 is also developed by committee, not widely adopted but influential in development of other languages.)

1960: COBOL runs on UNIVAC II and RCA 501.

1960: Control Data Corporation delivers its first product, a large scientific computer named the CDC 1604.

1960: First electronic switching central office becomes operational in Chicago.

1960: Removable disks first appear.

1961: AFIPS (American Federation of Information Processing Societies) forms.

1961: Multiprogramming runs on Stretch computer. Time-sharing runs at MIT on IBM 709 and 7090 computers by F. Corbato.

1961: IBM delivers the Stretch computer to Los Alamos. This transistorized computer with 64-bit data paths is the first to use eight-bit bytes; it remains operational until 1971.

1961: Jack Kelley and Robert Noyce of Fairchild Semiconductor develops the idea of a monolithic structure for integrated circuits. Noyce gets the patent.

1962: The first CDP examination was held in NEW YORK. George Abbot, of the AK-SAR-BEN Chapter in Omaha received CERTIFICATE # 1.

1962: NMAA elected to adopt a more progressive name to reflect the changing nature of information processing. DATA PROCESSING MANAGEMENT ASSOCIATION (DPMA) was founded.

1962: APL (A Programming Language) is developed by Ken Iverson, Harvard University and IBM.

1962: First general-purpose simulation languages are proposed: (1) SIMSCRIPT by the Rand Corporation, and (2) GPSS by IBM.

1962: IBM markets 1311 using removable disks.

1962: IBM's U.S.-based annual revenues from computer products reaches $1 billion and for the first time surpasses its other revenue.

1962 - Atlas computer from Univ Manchester, England, is first to have virtual memory and paging. Capable of 200k FLOPS.

Teletype ships Model 33 keyboard / punch-tape terminal.

1962: H. Ross Perot founds EDS (Electronic Data Systems) in Dallas, TX.

1963 - ASCII is the result of early efforts to develop standardization between various brands of computers. 'A Standard Code for Information Interchange'

1963: Control Data acquires Bendix Corp. computer division.

1963: Conversational graphics consoles are developed by General Motors (DAC-1) and MIT Lincoln Laboratories (Sketchpad), resulting in computer-aided design (CAD). Sketchpad uses the first light-pen, developed by Ivan Sutherland.

1963: DEC ships the first PDP-5 minicomputer.

1963: Charles Tandy buys Radio Shack Corp. -- for free!

1964: Control Data Corporation introduces the CDC 6000, which uses 60-bit words and parallel processing. CDC ships the 6600, the most powerful computer for several years. It was designed by Seymour Cray.

1964: BASIC (Beginners All-purpose Symbolic Instruction Language) is created by Tom Kurtz and John Kemeny of Dartmouth. First time-sharing BASIC program runs.

1964: Graphic tablet is developed by M.R. Davis and T.D. Ellis at Rand Corporation.

1964: Honeywell introduces the H-200 attacking IBM's installed base of 1400 systems.

1964: NCR introduces the 315/100.

1964: Douglas Engelbart, SRI, Automation Research Center, originates ideas for a number of modern computing concepts: hypertext, outline processor, video conference, the mouse, two-D editing, windows, cross-file editing, uniform syntax construction,

remote procedure protocols, mixed text and graphics files, and others.

1964: IBM produces first large scale, real-time, on-line reservation system - SABRE - for American Airlines.

1964: IBM coins the term "word processing".

1964 - April - IBM announces the System 360, an upward compatible, combination - scientific / business computer(by the mid-80's the IBM 360 will have generated over $100 billion in revenues).

1964 Paul Baron of the RAND Corporation, (America's foremost Cold-War Think Tank) makes public his proposal for a totally decentralized network - no central point of authority or control, "a network designed from the beginning to operate while in tatters."

(THE INTERNET IS CONCEIVED)

1965: CDC founds the Control Data Institute to provide computer-related education.

1965: Ken Olsen and Digital Equipment Corporation introduce the DEC PDP-8, first true mini computer.

1965: First computer science Ph.D. is granted to Richard L. Wexelblat at the University of Pennsylvania.

1965: IBM ships the first System 360, its first integrated circuit-based, or third generation, computer.

1966: Honeywell acquires Computer Control Company, a minicomputer manufacturer.

1966: Scientific Data Systems (SDS) introduces Sigma 7.

1966: Texas Instruments offers the first solid-state hand-held calculator.

1966 - National Science Foundation (NSF) cuts funding to universities for the development of (new) computers. Encourages the use of commercially available machines.

1967 - The NSF "Pierce Report" provides impetus for developing computer science curriculum for higher education.

1967 - Niklaus Wirth begins development of PASCAL language in Zurich, Switzerland.

1967 - Seven years after Fairchild introduced the integrated circuit,

the new third generation computers adopt IC technology.

1967: DEC introduces the PDP-10 computer.

1967: A.H. Bobeck at Bell Laboratories develops bubble memory.

1967: Burroughs ships the B3200.

1967: First issue of Computerworld is published.

1968 - Edward Dijkstra begins move against the 'jump' instruction in software. Movement to reliable software development is underway.

"GOTO Statement
Considered Harmful."

1968: Dendral, the first medical diagnostic medical program, is created by Joshua Lederberg at Stanford University.

1968: Univac introduces the 9400 computer.

1968: Integrated Electronics (Intel) Corp. is founded by Gordon Moore and Robert Noyce.

1968 - Arthur C. Clarke introduces HAL through the movie *"2001: A Space Odyssey"*.

1969 - Dennis Ritchie and Kenneth Thompson, Bell Labs, withdraw from multi-vendor 'Multics' operating system program and begin work on a 'single user' operating systems. They call it UNIX.

1969: Edson deCastro leaves DEC to start Data General Corp. and introduces the Nova, the first 16-bit minicomputer.

1969: First International Joint Conference on Artificial Intelligence is held.

1969: IBM unbundles hardware and software; introduces a minicomputer line, System/3.

1969: Lockheed Electronics ships the MAC-16.

1969: PASCAL compiler is written by Nicklaus Wirth and installed on the CDC 6400.

1969 - Intel announces the 1KB RAM chip - highest capacity ever.

1969 - Xerox opens Palo Alto Research Center (PARC)

1969

Peace, Love, Music and

DoD commissions ARPAnet for research into networking also in 1969

Bill Gates and Paul Allen, calling themselves *Lakeside Programming Club*, sign an agreement with Computer Center Corporation to report bugs in PDP-10 software in exchange for free computer time.

1970: Computer Logic Systems ships SLS-18.

1970: DEC ships its first 16-bit minicomputer, the PDP-11/20.

1970: Data General ships SuperNova.

1970: First ACM Computer Chess tournament is held.

1970: Honeywell acquires General Electric's computer operations.

1970: IBM ships its first System 370, a fourth generation, computer.

1970: Xerox Data Systems introduces the CF-16A.

1970

Gilbert Hyatt files patent application for "Single Chip Integrated Circuit Computer Architecture" the first basic patent on the microprocessor.

First PASCAL compiler becomes operational.

Information Sciences contacts Gates and Allen, offering PDP-10 computer time for programming expertise.

Frederico Faggin, Intel, begins work on circuit design for 4004 microprocessor.

1971 - Marcian Ted Hoff, Intel, delivers the 4004 for ETI, a Japanese calculator company.

4-bit bus –
108 KHz,60,000 operations/sec, 2300 transistors,640 bytes addressable, US $200

Documentation manuals are written by Adam Osborne.

Alan Shugart, IBM, delivers practical use of the 8" floppy disk on the Displaywriter dedicated word processor.

Steve Wozniak and Bill Fernandez build a computer from rejected parts - call it 'the Cream Soda Computer'.

Wang Labs introduces the Wang 1200 word processor.

15 nodes on ARPANET

(THE INTERNET BEGINS)

1971: Computer Automation introduces the Alpha-16.

1971: IBM introduces the 370/135 and 370/195 mainframe computers.

1971: Floppy disks are introduced to load the IBM 370 microcode.

1971: Intel Corporation announces the first microprocessor, the Intel 4004, developed by a team headed by Marcian E. Hoff.

1971: John Blankenbaker builds the first personal computer, the Kenbak I.

1971: NCR introduces the Century 50.

1971: Sperry-Rand takes over the RCA computer product line.

1972: Cray Research is founded.

1972: First electronic pocket calculator is developed by Jack Kilby, Jerry Merryman, and Jim VanTassel of Texas Instruments.

1972: Gary Kildall at Naval Postgraduate School writes PL/1, the first programming language for the Intel 4004 microprocessor.

1972: Prime Computer is founded.

1972

Intel develops the 8008 chip for Computer Terminal Corp

8-bit bus

108 KHz,

3500 transistors,

16K bytes address space

Atari is founded by Nolan Bushnell - ships the first commercial video game - PONG

Bill Gates and Paul Allen form the 'Traf-O-Data Company' after developing an 8008-based turnkey system for recording automobile traffic flow on highways.

First 5.25 inch floppies appear

Edward Roberts, William Yates & Jim Bybee, Micro Instrumentation & Telemetry Systems, deliver the MITS 816 to computer hobby enthusiasts no display, no keyboard, no storage

1973

Based on the Intel 8008, the French built Micral, first non-kit microcomputer, is advertised unsuccessfully in the U.S. first reference of "microcomputer" in print

Donald Knuth promises 12 volumes of *"The Art of Programming."*

First three become the 'bible' of software engineering.

Univ College of London & Royal Radar of Norway are first international ARPANET nodes

Bob Metcalfe's Harvard Thesis outlines the idea for Ethernet

Xerox builds the Alto workstation at PARC. Uses Smalltalk language, a mouse & Ethernet. Less than 2000 are built.

Stephen Wozniak joins Hewlett-Packard

Gary Kildall begins consulting work at Intel.

1973: First National Computer Conference (NCC) is held in New York City.

1973: IBM settles a lawsuit by Control Data, selling Service Bureau Corporation (SBC) to Control Data.

1973: PROLOG language is developed by Alain Comerauer at the University of Marseilles-Luminy, France.

1973: R2E markets the MICRAL, the first microcomputer in France.

1973: Winchester disk drives are first introduced by IBM, who uses the term as a code name for its Model 3340 direct-access storage device.

1974: Digital Equipment enters the Fortune 500 ranking of the largest industrial companies.

1974: DPMA helps establish the "INSTITUTE for the CERTIFICATION of COMPUTER PROFESSIONALS". This organization was formed to stimulate industry acceptance of the examinations. The ICCP begins administering DPMA's CDP program.

1974: Intel introduces the 8080, an 8 bit microprocessor that will be used in numerous personal computers.

8-bit bus

2 MHz,

6000 transistors,

64K bytes address space

1974: Zilog is formed.

1975: Homebrew Computer Club, considered the first personal computer users group, is formed.

1975: MITS introduces the Altair personal computer, named after a Star Trek episode, A Voyage to Altair. The kit cost $397 for a 256 byte computer. The I/O consisted of switches and lights. It was designed by Ed Roberts and Bill Yates.

1975: Microsoft is founded after Bill Gates and Paul Allen adapt and sell BASIC to MITS for the Altair PC.

1975: The first computer store opens in Santa Monica, CA.

1975: Xerox withdraws from the mainframe computer industry.

1975 has a few surprises

IBM announces the 5100 'educational' computer

BASIC,

16KB Ram,

tape storage,

5" screen

Price: $9000

Weight: 55 pounds - sales are disappointing

Cray I Supercomputer announced by Seymour Cray

First issue of *Byte* magazine is published.

Steve Dompier uses his Altair and a radio to play "Fool on the Hill" & "Daisy" at the Homebrew Computer Club.

Gates and Allen change company name to Micro-Soft

1976: First fault-tolerant computer, the T/16, is introduced by Tandem.

1976: MYCIN, an expert system to diagnose and treat infectious blood diseases, is developed at Stanford University by E. Shortliffe.

1976: NEC System 800 and 900 general-purpose mainframes are introduced.

1976: Seymour Cray engineers and delivers Cray 1 with 200,000 freon-cooled ICs and 100 million floating point operations per second (MFLOP) performance.

1976: Super minicomputers are introduced by Perkin-Elmer and Gould SEL.

1976: Zilog Z-80 chip is introduced.

1977: Steve & Steve name a computer after a piece of fruit. Jobs ,Wozniak

Apple Computer is founded and introduces the Apple II personal computer.

1977: Apple, Commodore, and Tandy begin selling personal computers.

1977: DEC introduces its first 32-bit super minicomputer, the VAX-11/780.

1977: Datapoint introduces ARC system, the first local area network.

1977: First ComputerLand franchise store opens in Morristown, NJ under the name Computer Shack.

1977: Tradename 'Microsoft' is registered

1978: SPRINT business service is inaugurated.

1978: Texas Instruments introduces the Speak-and-Spell educational toy featuring digital speech synthesis.

Total computers in use in the U.S. exceed a half million units.

1978: The first COMDEX trade show is held.

1978 - Apple licenses BASIC from Microsoft as *Applesoft*

Microsoft sales reach $1 million for the year.

1978 - Daniel Bricklin and Bob Frankston introduce VISICALC - a new concept for application computing.

Scott Adams founds Adventure International

1978

Intel releases the 8086 chip

16-bit registers,

16-bit bus

29,000 transistors,

1M bytes address space

$360

follows with the 8088 as a stepping stone to 8086 16-bit internal, 8-bit to external devices

Moore's Law

The density of transistors on a chip will double every 18 months, thus increasing the price performance of compute power by a factor of two every 1 1/2 years.

Gordon Moore, Co-Founder, INTEL Corp.

1979: Ada language is developed by a team at CII-Honeywell Bull (France) directed by Jean Ichbiah.

1979: The Source and CompuServe Information Services go on-line.

1979: VisiCalc, the first electronic spreadsheet software, is shown at the West Coast Computer Faire.

1979: Wordstar, one of the best-selling word processing programs for PCs, is released by Micropro (now called Wordstar International).

1979 - Taito introduces Space Invaders in Japan.

1980: Control Data Corporation introduces the Cyber 205 supercomputer.

1980: First issue of InfoWorld is published.

1980: Microsoft licenses UNIX operating system from Bell Laboratories and introduces its XENIX adaptation.

1980 - Tim Patterson begins writing a disk-based operating system for use with Seattle Computer Products(SCP) 8086-based computer.

Paul Allen contacts SCP asking for rights to sell Patterson's DOS to an 'unnamed client.'

Microsoft pays less than $100,000 for the rights.

1980 - Alan Shugart, after leaving IBM, introduces the Winchester hard drive for PCs. This changes everything. and

in exchange for MSC carrying the development costs.

IBM underestimates the revolution!!

1980: Total computers in use in the U.S. exceed one million units.

1981: Commodore introduces the VIC-20 home computer, which sells over one million units.

1981: IBM enters the PC arena with the IBM PC.

It is supported by the DOS operating system

from Microsoft Corporation, under an agreement that gives Microsoft all profits

IBM 5150 Personal Computer (PC)

4.77 MHz Intel 8088 CPU

4KB RAM,

40KB ROM

5.25 " floppy drive,

PC-DOS 1.0 (MS-DOS)

$3000 base price

$6000 fully expanded

Wall Street ad from Apple: *"Welcome IBM... Seriously!"*

Tandy President, John Roach, "I don't think IBM's entry into the microcomputer field is that significant."

Microsoft begins work on GUI a 'Graphical User Interface'.

Apple Computer prohibits mail-order sales - claiming, "no provisions for customer education or support services."

Osborne Computer Co. begins marketing the first fully self-contained portable computer. (bankrupt in two years)

College professor, James Clark, founds Silicon Graphics

1981: Osborne Computer introduces the Osborne 1, the first portable computer.

THE INFORMATION AGE is Announced

1982

John Naisbitt, Megatrends - "The information age will collapse the information 'float'.

#1 point of the 5 key points of the information age:

"The Information Society is an economic reality, not an intellectual abstraction."

1982

Justice Department throws out 13 year old antitrust lawsuit against

IBM.

Disney's TRON - special effects are computer generated.

Intel releases the 80286 chip

16-bit registers,

16-bit bus

134,000 transistors,

16M bytes address space

$360

Rod Canion, Jim Harris & Bill Murto, senior managers at Texas Instruments, leave to found Compaq Computer

and . . .

Commodore Super VIC

TI 99/4

Toshiba T-100

Radio Shack TRS-Model 16

Casio FX-9000P IBM-PC XT Epson KX-1

Sharp PC-1500 NEC 5200 Sinclair ZX81 Altos 8600 TRS Pocket Computer Atari 800 Astrovision ZGrass-32 IBM AT

Kaycomp II Coleco Vision Olivetti M20 Wang Professional Computer Victor 9000 Timex Sinclair 1000 PC-Clones Apple II Epson HX-20 Handheld Hitachi 16000 Digital Equipment Corporation Rainbow 100

Franklin Ace 1000 SordM23P Aval AVC-777J2 LISA

Apple is first PC company to hit $1 billion in sales

1982: AT&T agrees to give up 22 Bell System companies in settling a 13-year-old lawsuit brought by the Justice Department.

1982: Compaq Computer incorporates.

1982: Sun Microsystems is founded.

1982: Microsoft licenses MS-DOS to 50 microcomputer manufacturers in the first 16 months of availability.

1982: *TIME* magazine features the 'PC' as "Man of the Year".

1983: Compaq ships its first computer in January and sells $111M, the greatest first-year sales in the history of American business.

1983: Cray 2 computer introduced with one billion FLOPs (floating point operations per second) performance rating.

1983: Mitch Kapor introduces LOTUS 1-2-3 Lotus 1-2-3 replaces VisiCalc as the spreadsheet software of choice for microcomputers.

1983: NEC announces the SX-1 and SX-2 supercomputers.

1983: Total computers in use in the U.S. exceed ten million units.

US Dept of Defense announces the Ada language after five years of successive refinements - the high-order language is widely criticized for its complexity.

ARPANET spins off MILNET for Defense Network

Apple produces the 1,000,000th Apple II

IBM & Microsoft begin joint development of OS/2

Wang announces single in-line memory module (SIMM)

AT&T Bell Labs designs C++

MS Windows formally announced - IBM not interested, has Top View plans

1984

Steve Jobs delivers the MAC after "seeing the light" at Xerox PARC. The mouse and icon come to the people.

Appleworks - one of the first integrated office packages written by Rupert Lissner.

2,000,000 Apple II sold

3rd and final demo of Windows to IBM - still no interest

1000 hosts on the ARPANET

1984: Apple introduces the Macintosh computer.

1984: IBM introduces the PC AT (Advanced Technology). IBM merges with Rolm Corp., which becomes a telecommunications subsidiary.

1984: The Tandy 1000 personal computer becomes the #1 selling IBM PC-compatible in its first year.

1985: IBM delivers the new 3090 Sierra systems.

1985: Aldus introduces PageMaker for the Macintosh and starts the desktop publishing era.

IBM discontinues PC jr

Computer Crackers come to forefront when"414 Hackers" of Milwaukee break into the Los Alamos Laboratory computer system.

Steve Jobs is unimpressed with preview of MS Excel, prefers Lotus Jazz

Apple Computer reports first quarterly loss

Jobs 'leaves' Apple Computer - forms NeXT Inc.

Ted Waitt founds Gateway 2000 in Sioux City, IA

Windows 1.0 ships (November)

IBM announces Token Ring Architecture

Microsoft purchases all rights to DOS from SCP - $925,000

Nintendo is introduced to the U.S. market

1985

Intel announces the 80386 chip

32-bit registers,

32-bit bus

16-MHz

-275,000 transistors,

4Gig bytes address space

-$299 in quantity

1986: Burroughs merges with Sperry to form Unisys Corporation, second only to IBM in computer revenues.

1986: Compaq makes the Fortune 500 list. Introduces its first Intel 80386-based PC.

1986: Computerworld publishes its 1,000th issue on November 3.

1986: HP introduces its Spectrum line of reduced instruction set computers (RISC).

1986: Tandy has over 7300 retail outlets including more than 4800 company-owned Radio Shack stores in the U.S.

1986: The number of computers in the U.S. exceeds 30 million.

1987: IBM introduces its PS/2 family and ships over 1 million units by year end.

1987: Cray Research introduces the Cray 2S which is 40% faster than the Cray 2.

1987: ETA Systems introduces its ETA-10 family of supercomputers.

1987: Sun Microsystems introduces its first workstation based on a RISC microprocessor.

1987: Apple introduces the Macintosh II and Macintosh SE and HyperCard.

1987: IBM introduces its Systems Applications Architecture (SAA).

1987: DEC introduces Vaxstation 2000 workstation computer, and the MicroVAX 3500 and 3600.

1987: Aldus introduces PageMaker for the IBM PC and compatible computers.

1987: Compaq reaches a billion dollar in sales in its fifth year of operation.

1987: Conner Peripherals beats Compaq's first year sales record: $113M vs $111M.

1987: Computer Associates acquires UCCEL in the largest ever software acquisition ($780M).

1987: IBM invests in Steve Chens Supercomputer Systems, Inc.

1987: Apple spins off its application software business as a separate company and names it Claris.

1987: Texas Instruments introduces the first AI microprocessor chip.

1988: DEC introduces VAXstation 8000.

1988: Cray Research introduces the Cray Y-MP, a $20M supercomputer.

1988: IBM introduces a new mainframe computer operating system called MVS/ESA.

1988: IBM announces its long awaited Silverlake mid-range computers called AS/400.

1988: Motorola announces the 88000, a RISC microprocessor.

1988: The first graphics supercomputers are announced by Apollo, Ardent and Stellar. These computers are aimed at 3D graphics applications.

1988: The first PS/2-compatible computers are announced by Tandy, Dell Computer and others.

1988: Unisys introduces the 2200/400 family to replace its mid-range 1100 series.

1988: AT&T announces plan to acquire 20% of Sun Microsystems, and that Sun will help AT&T develop the next version of UNIX.

1988: In response to the AT&T-Sun cooperation, IBM, DEC, HP, Apollo and several other major computer companies form the Open Software Foundation to set a UNIX counterstandard.

1988: Sun Microsystems surpasses the $1 billion sales mark, and introduces 80386-based workstations.

1988: IBM and Sears joint videotex venture starts operation under the PRODIGY name.

1988: Sematech picks Austin, TX as its headquarters and the consortium will be headed by Robert Noyce.

1988: A consortium of PC companies led by Compaq introduces the EISA counter standard to IBM's PS/2 MicroChannel bus.

1988: IBM introduces the ES/3090 S series mainframe computer.

1988: IBM wins a $3.6B contract to build the next generation air traffic control system.

1988: Unisys acquires Convergent Technologies for $350M.

1988: Computer Associates acquires Applied Data Research for $170M from Ameritech.

1988: Next unveils its innovative workstation computer which is the first computer using erasable optical disks as the primary mass storage device. IBM license Next's graphics user interface.

1988:

A nondestructive worm spreads via the Internet network and brings several thousand computers to their knees.

1988

Compaq Computer reports sales of $1.2 billion - quickest a company has ever reached that mark

Apple sues Microsoft & Hewlett Packard - MAC OS issues

Ashton-Tate sues Fox - Dbase language

DEC begins development of 64-bit, 150-MHz alpha chip

W.H. Sim forms Creative Labs, Inc.

HP introduces the DeskJet inkjet printer - $1000

1989

Intel announces the 80486 chip combines

386 & 387 math coprocessor & cache

1.2 million transistors

$900

NeXT, Inc. ships its first machine

Creative Labs releases 8-bit mono Sound Blaster card

Personal Computer Memory Card International Association (PCMCIA) is formed to develop standards for PCs.

LOTUS Development quote: *"We don't see Windows as a long term graphical interface for business."*

Apple announces a new font standard - it will become "TrueType"

First relays between a commercial e-mail carrier and the Internet - MCI/CRNI & Compuserve/OSU

100,000 hosts on Internet

1989: Solbourne Computer introduces the first Sun 4-compatible computer.

1989: DEC announces a workstation using Mips Computer's RISC microprocessor.

1989: Microsoft buys a 20% stake in Santa Cruz Operation, a major UNIX software developer.

1989: Intel announces the 80486 microprocessor and the I860 RISC/coprocessor chip. Both chips have over one million transistors.

1989: Hewlett-Packard acquires Apollo for $476M.

1989: Sun Microsystems introduces its SPARCstation, a low-end RISC workstation with an entry price of only $9,000.

1989: Control Data discontinues its ETA supercomputer subsidiary.

1989: IBM announces the Officevision software using the SAA protocol, which runs on PS/2s, PS/2 LANs, AS/400 and mainframe computers.

1989: Cray restructures itself into two companies: Cray Research which continues with its current business and Cray Computer Corp. headed by Seymour Cray, which will develop a gallium arsenide-based supercomputer.

1989: Next sells a 16.6% share to Canon for $100M.

1989: Seagate buys Control Data's Imprimis disk drive subsidiary for $450M.

1989: Computer Associates acquires Cullinet for $353M.

1989: Prime Computer agrees to be bought by a J.H. Whitney-formed company, ending a long and acrimonious takeover battle by MAI Basic.

1989: Apple introduces its long awaited portable Macintosh.

1989:

The worldwide number of computers in use surpasses 100M units.

1989: Poqet announces the first pocket sized MS-DOS compatible computer.

1989: Grid introduces a laptop computer with a touch sensitive pad that recognizes handwriting--the GridPad.

1989: The battery-powered notebook computer becomes a full function computer including hard and floppy disk with the arrival of Compaq's LTE and LTE/286.

1989: Digital Equipment extends the VAX-family into the mainframe arena with the VAX 9000.

1989: The first EISA-based personal computers arrive.

1989: The first 80486-based computers are introduced.

1989: Dun & Bradstreet acquires MSA in a major software acquisition worth $333M.

1990: Motorola introduces the 68040 microprocessor.

1990: IBM announces its RISC Station 6000 family of high performance workstations.

1990: Digital Equipment introduces a fault-tolerant VAX computer.

1990: Cray Research unveils an entry-level supercomputer, the Y-MP2E, with a starting price of $2.2M.

1990: Microsoft introduces Windows 3.0.

1990: Lotus wins its look and feel suit against Paperback Software's spreadsheet program.

1990: IBM ships the PS/1, a computer for consumers and home

offices.

1990: IBM announces the System 390 (code name Summit), its mainframe computer for the 1990s.

1990: Microsoft's fiscal year revenue ending 6/30/90 exceeds $1B.

1990: NCR abandons its proprietary mainframes in favor of systems based on single or multiple Intel 486 and successor microprocessors.

1990: Apple introduces its low-end Macintoshes: The Classic, LC and IISI.

1990: Intel launches a parallel supercomputer using over 500 860 RISC microprocessors.

1990: Sun Microsystems brings out the SPARCstation 2.

1990: Microsoft along with IBM, Tandy, AT&T and others announced hardware and software specifications for multimedia platforms.

1990: The first SPARC compatible workstations are introduced.

1990

ARPANET ceases to exist - NSF assumes funding

Microsoft releases Windows 3.0 $3 million 1st day announcement for $10,000,000 plan

Microsoft annual sales reach $1 billion,

first personal software company to do so.

Gilbert Hyatt is granted a basic patent for the microprocessor, 20 years after his first application for patent.

IBM & Microsoft end cooperative work agreement

1991

Compaq reports billion dollar quarter

Apple & IBM sign technology sharing agreement

Brad Silverberg, MSC VP, "DOS is here forever."

Commercial Internet Exchange (CIX) is formed

Wide Area Information Service (WAIS) - Kahle

Gopher - Lindner & McCahill

PGP (Pretty Good Privacy) - Zimmerman

World Wide Web (WWW)

Tim Berners-Lee CERN releases the first Web server

Business spending on computing exceeds spending for industrial, mining, farming and construction equipment.

1991: Go Corp. releases PenPoint, an operating system for pen-based computers.

1991: Advanced Micro Devices announces its AMD 386 microprocessor to compete with Intel's 386 chips.

1991: Notebook PCs are introduced by most PC vendors.

1991: HP unveils its RISC-based 9000 Series 700 workstations with exceptional price-performance.

1991: Compaq leads a group of 21 companies to launch the Advanced Computing Environment (ACE) to establish a new standard for high-end PCs and workstations.

1991: The Federal Trade Commission launches an investigation into Microsoft's business practices.

1991: Intel introduces the 486SX, a lower priced 486 chip.

1991: NCR agrees to be acquired by AT&T in a deal valued at $7.4B.

1991: Apple releases the System 7.0 operating system for Macintosh.

1991: Wang will resell IBM's PS/2, RS/6000 and minicomputers. IBM will invest $100M in Wang.

1991: Microsoft rolls out DOS 5.0 with great success.

1991: Major changes among PC dealers as:

ComputerLand acquires Nynex's computer stores,

CompuCom acquires Computer Factory,

ValCom and Inacomp merge;

JWP buys Businessland;

Intelligent Electronics acquires BizMart.

1991: Borland buys Ashton-Tate for $440M.

1991: SunSoft, a Sun Microsystems subsidiary, announces Solaris which is a UNIX operating system for SPARC workstations and 386/486 PCs.

1991:- The Bell companies receive permission to enter the on-line information services market.

1991: Apple and IBM sign a historic deal--including two joint ventures: Kaleida will develop multimedia products, Taligent will develop object-oriented operating software.

1991: Apple rolls out its PowerBook notebook and Quadra Macintosh PCs.

1991: Wavetracer introduces its Zephyr massively parallel computer system with up to 8192 processors.

1991: IBM reorganizes itself into more autonomous business units and several divisions become wholly-owned subsidiaries.

1991: AT&T/NCR agrees to acquire Teradata for $520M.

1991: Many major computer companies have quarterly or full-year loses including Compaq, DEC, IBM, Lotus and Unisys, primarily due to work force reduction costs.

1991: The first general purpose pen-based notebook computers are introduced.

1991: IBM has its first revenue decline in 45 years.

1992

IBM reports first ever year end loss $564 million on sales of $64.8 billion

Intel announces 'clock doubler' - debuts 486DX2

IBM announces ThinkPad laptop computer

Apple & Sharp agree to co-develop the Personal Digital Assistant (PDA)

1,000,000 hosts on the net

1992

Solomon Waters of Altadena, CA, a six year old first grader, comes home from school and reports that he has written on "a machine that looks like a computer but has no TV screen." His mother asks if it was a "typewriter?"

"Yeah! That's what it was!" is his reply. L.A.Times

1992: IBM invests $100M in Groupe Bull.

1992: Silicon Graphics buys Mips Computer in a $400M stock swap.

1992: IBM releases OS/2 Version 2.0 and ships over 1M units.

1992: Microsoft introduces Windows 3.1 and ships nearly 10M units.

1992: The core of Apple's lawsuit versus Microsoft Windows is dismissed.

1992: Sun Microsystems launches a new generation of SPARC computers--the SPARCstation 10 family.

1992: Compaq announces several new lines of PCs and becomes a price trend setter. Its low-price strategy is very successful.

1992: Ken Olsen resigns from Digital Equipment after 25 years at the helm.

1992: Sears and IBM forms a new venture, named Advantis, to compete in the value added network service market.

1992: Wang Laboratories files for Chapter 11 bankruptcy protection.

1992: IBM makes the IBM PC Co. a subsidiary.

1992: IBM follows Compaq's strategy and introduces aggressively priced PCs--also with good success.

1992: Compaq enters the Japanese market with aggressively priced PCs--as much as 50% lower than Japanese PC prices.

1992: Digital Equipment announces its next generation computer

architecture--the RISC-based Alpha.

1992: Microsoft introduces Windows for Workgroups..

1992: Intel says its next microprocessor will be called Pentium instead of 586.

1992: Hewlett-Packard ships the LaserJet 4, a 600 by 600 dots per inch resolution laser printer.

1992: Novell to acquire UNIX Systems Laboratory, including Univel, from AT&T for $350M.

1993: IBM reports its worst year in history with a loss of $4.97B on revenues of $64.5B.

1993: IBM chairman John Akers resigns and after the most executive search publicity ever, Louis Gerstner becomes the new chairman & CEO.

1993: General Magic, an Apple spin-off, debuts Telescripts, a communications-intensive operating system for PDAs.

1993: Next sells its hardware business to Canon and will concentrate its effort on the Nextstep software business.

1993: Novell unveils NetWare 4.0.

1993: IBM introduces the F series of the AS/400.

1993: Lotus announces Notes 3.0.

1993: Motorola start shipping the first PowerPC microprocessor.

1993: IBM's storage division, Adstar, becomes a subsidiary.

1993: - Microsoft unveils Windows NT.

1993: Pentium-based systems start shipping.

60-MHz Pentium

64-bit bus

32-bit registers

3.2 million transistors

$878

MS-DOS 6.0 sells 1 million retail copies in first 40 days.

Gateway ships # 1,000,000

InterNIC created by NSF to provide specific Internet services

White House & United Nations come on-line

Mosaic (NCSA) takes off - co-developer, Marc Andreesen

WWW has 341,634% annual growth rate

Microsoft ships Windows NT & reports a $1 billion quarter

1993: Microsoft outlines the Plug and Play and Microsoft at Work (MAW) initiatives

1993: EPA's Energy Star Initiative is unveiled and most PC vendors support the program with announcements of energy efficient PCs.

1993: Apple ships the Newton MessagePad--its first Personal Digital Assistant.

1993: AT&T announces it will acquire McCaw Cellular for $12.6B.

1993: Compaq introduces the Presario, a PC family targeted for the home market.

1993: FTC ends its probe of Microsoft without any actions, but the Antitrust Division of the Department of Justice will launch its investigation.

1993: IBM debuts its first workstations based on the PowerPC chip.

1993: Novell transfer the UNIX trademark to X/Open and X/Open will certify that an operating system is UNIX compliant.

1993: IBM announces OS/2 for Windows, which upgrades the Windows environment to OS/2.

1993: Sun Microsystems license NextStep and makes a $10M investment in Next.

1993: IBM say it will sell its Federal Systems division ($2.2B in yearly revenue) to Loral for $1.6B.

1994: John Sculley leaves Apple after 10 years at the helm.

1994

US District Court rules Microsoft violated patents held by Stac Electronics for disk compression. Ordered to remove or replace the technology, buys $40 mil stock + pays $43 mil royalties.

Microsoft settles lawsuit alleging monopolistic licensing.

Dr. Thomas R. Nicely, Lynchberg College, notices the Intel Pentium sometimes produces 'reduced accuracy' results. Intel confirms flaw, Andy Grove offers apologies to 2 million Pentium owners, later offers to replace all defective floating-point chips. Remedy costs Intel ~$475 million.

1994: Apple enters the on-line service market by announcing eWorld.

1994: HP becomes a Taligent partner and buys 15% from Apple and IBM.

1994: MCI invests $1.3B in Nextel Communications, a wireless service provider.

1994: Macintoshes using the PowerPC start shipping.

1994: Intel introduces the 486DX4 clock-tripling microprocessor.

1994: Aldus and Adobe agree to merge in a transaction worth $525M and will form a $0.5B+ software company.

1994: Novell says it will acquire WordPerfect for $1.14B and will buy Borland's Quattro Pro for $145M.

1995

Apple ships 1 millionth Power Mac.

IBM announces 1 million copies OS/2.

Windows 95 is released with no small fanfare 1 million copies sold through retail in first 4 days.

NSFNET reverts back to research network - commercial providers begin carrying the backbone Internet traffic.

Compuserve, AOL and Prodigy begin Internet access.

Registration of Domain names - no longer free - now $50.

August 9 - Netscape becomes 3rd largest NASDAQ IPO offering ever - Jim Clark, Mark Andreesen, James Barksdale.

Intel Pentium Pro at

150-200 MHz

$974 - $1682

9,000,000 hosts connected WWW

Moore's Second Law

The cost of building chip fabrication plants will continue to increase (and the return on investment to decrease) until it becomes fiscally untenable to build new plants.

1995 - Gordon Moore, Co-Founder, INTEL Corp.

1996

Microsoft ships 30 millionth copy of Windows 95.

Digital announces 433 MHz and 500 MHz Alpha processors.

Intel delivers the $55 million "Teraflop" machine to Sandia National Labs - Dept of Energy.

7,264 Pentium 200's in parallel

July 1996
12,880,000 hosts connected to the Internet

1996: October - DPMA members elect to change the Association's to the "ASSOCIATION of INFORMATION TECHNOLOGY PROFESSIONALS"

As before, the changing nature of the industry seemed to dictate a more inclusive and progressive name.

January 1997

Intel announces 200-MHz Pentium MMX

64-bit bus

32-bit registers

32 Kb on-board cache

4.5 million transistors

$550 ea. qty 1000

January 1, 1997

DPMA officially becomes the "ASSOCIATION of INFORMATION TECHNOLOGY PROFESSIONALS"

Acknowledgments to:

IEEE Computer Society '50th anniversary web pages' - www.computer.org/50/history/index.html

Mr. Jaques Neirynck, Swiss Federal Institute of Technology - sgline.epfl.ch/mbone/neirynck_start.html

History of Telecommunications - www-stall.rz.fht-esslingen.de/telehistory/comp30.html#1944

Ken Polsson, Chronology of Events in the History of Microcomputers - www.islandnet.com/~kpolsson/comphist.htm

PBS ONLINE - Triumph of the Nerds - web-cr02.org/nerds

. . . and other WWW & industry journal sources

Timeline and display courtesy AITP- AK-SAR-BEN Chapter - Omaha, NE

April 26, 27, 28, 1999
Monday, Tuesday, Wednesday
Holiday Inn Convention Center
I-80 & 72nd Street
Omaha, Nebraska
Show Hot Line: 1-800-996-3233

Calendar | Overview | Home | aitp | Management | Copyright | Email

Comprehensive Computer Catalogue ©

COMPUTERS MADE IN THE SIXTIES

Last updated 4 Dec 1997 5:01

Quick Links

Pre 1960 | 1960 | 1961 | 1962 | 1963 | 1964 | 1965 | 1966 | 1967 | 1968 | 1969 | After 1969 | Dates Unknown | Organisations | References

Builder	Machine Name	Origin	Date	Re
RCA	300	USA	1960	B
ICT	1101	GB	1960	P
Philco	3000	USA	1960	B
	AN/MYK		1960	B
	AN/TYK-4V COMPAC		1960	B
	AN/USQ-20		1960	A
Litton	C7000	USA	1960	B
SEA	CAB 500	FR	1960	p
U of Pisa	CEP	IT	1960	C
	CSC-160	AU	1960	B
Burroughs	D 107	USA	1960	B
Burroughs	D 825	USA	1960	B
RCA	DIP	USA	1960	E
EMI	EMIDEC 1100	GB	1960	P
Bull	Gamma 60	FR	1960	D
Danish Academy	GIER	DK	1960	
SNE	KL 901	FR	1960	C
Librascope	Libratrol 1000	USA	1960	B
	Mobidic 7A		1960	T
	Mobidic B	USA	1960	B
	Mobidic C		1960	T
	Mobidic D		1960	T
Philips	PASCAL	NL	1960	p
Fujitsu	PC-2	JP	1960	A
DEC	PDP-3	USA	1960	E
Ferranti	Sirius	GB	1960	S
SMI	SNOCOM	AU	1960	T
Philips	STEVIN	NL	1960	p
	Sylvania 59400		1960	B
	Target Intercept		1960	A

Tata Institute	TIFRAC	IN	1960	D
	Westinghouse Airborne		1960	B
Zuse	Z23	DE	1960	P
ELWRO	ZAM 2	PL	1960	A
CDC	1604	USA	Jan 1960	D
CDC	1604A	USA	Jan-60	P
Librascope	L-3000	USA	Jan-60	P
Sperry Rand Univac	Solid State 80/90 II	USA	Jan-60	P
Bunker-Ramo	TRW-300	USA	Jan-60	G
Clary Computer	DE 60	USA	Feb-60	B
Philco	2000-211	USA	Mar-60	B
Mitsubishi	Melcom 1101F	JP	Mar-60	P
Sperry Rand Univac	LARC	USA	May-60	E
Librascope	Libratrol XI	USA	May-60	B
Monroe	Monrobot XI	USA	May-60	B
IBM	7070	USA	Jun-60	B
CDC	160	USA	Jul-60	P
IBM	1401 (Card)	USA	Sep-60	E
IBM	1401 (Mag Tape)	USA	Sep-60	E
	AN/FSQ-31		Sep-60	B
	AN/FSQ-32		Sep-60	B
Burroughs	B 251	USA	Sep-60	P
Honeywell	H290	USA	Sep-60	P
Brookhaven National Lab	Merlin	USA	Sep-60	B
Royal Precision	RPC 9000	USA	Sep-60	P
AEI	1010	GB	Oct-60	P
IBM	1620	USA	Oct-60	P
Olivetti	ELEA 9003	IT	Oct-60	P
GE	210	USA	Nov-60	P
DEC	PDP-1 (M.T.)	USA	Nov-60	B
DEC	PDP-1 (P.T.)	USA	Nov-60	B
Royal Precision	RPC 4000	USA	Nov-60	P
Raytheon	250	USA	Dec-60	P
Honeywell Datamatic	800	USA	Dec-60	B
General Mills	AD/ECW-57	USA	Dec-60	B
Packard Bell	PB 250	USA	Dec-60	B
Bunker-Ramo	TRW-330	USA	Dec-60	G

1961

RCA	110	USA	1961	P
Sperry Rand Univac	1000	USA	1961	B
Sperry Rand Univac	1020	USA	1961	B
Sperry Rand Univac	1206	USA	1961	B
	AN/TYK-7V INFORMER		1961	A
Ferranti	Apollo	GB	1961	p
ITT	Bank Loan Process	USA	1961	B
	Citac 210B		1961	P
EMI	EMIDEC 2400	GB	1961	S
Bendix	G21	USA	1961	B
Bull	Gamma 500	FR	1961	P
	George II		1961	B
U of Chicago	MANIAC III	USA	1961	B

Sperry Rand Univac	NTDS	USA	1961	
	Oklahoma University		1961	B
	Serel 1001		1961	P
Marconi	TAC	GB	1961	A
	UMC1	PL	1961	B
	ZAM 21	PL	1961	A
	ZAM-2beta	PL	1961	E
ELWRO	APSAC	USA	Jan-61	B
General Mills	Basicpac	USA	Jan-61	A
Philco	301	USA	Feb-61	P
RCA	400	USA	Mar-61	P
Bunker-Ramo	803	GB	Mar-61	P
Elliott	BRLESC	USA	Mar-61	B
	NEAC 2101	JP	Mar-61	P
NEC	NEAC 2205	JP	Mar-61	P
NEC	225	USA	Apr-61	P
GE	255	USA	Apr-61	P
GE	310	USA	Apr-61	P
NCR	G20	USA	Apr-61	P
Bendix	390	USA	May-61	B
NCR	7030 aka STRETCH	USA	May-61	B
IBM	DDP-19	USA	May-61	B
CCC	DDP-25	USA	May-61	P
CCC	290	USA	Jun-61	B
Honeywell	HITAC 201	JP	Jun-61	P
Hitachi	Recomp III	USA	Jun-61	B
Autonetics	160A	USA	Jul-61	B
CDC	AD/ECS 37		Jul-61	P
General Mills	136	USA	Aug-61	P
CDC	924	USA	Aug-61	B
CDC	PC-2	JP	Aug-61	A
U of Tokyo	RW 530	USA	Aug-61	B
Ramo-Wooldridge	601	USA	Sep-61	D
RCA	7080	USA	Sep-61	P
IBM	B 250	USA	Sep-61	P
Burroughs	CE 102		Sep-61	P
	FACOM-222	JP	Sep-61	P
Fujitsu	MADIC IIA	JP	Sep-61	P
Matsushita	130	USA	Oct-61	P
Bunker-Ramo	1410	USA	Nov-61	E
IBM	HIPAC 103	JP	Nov-61	P
Hitachi	490	USA	Dec-61	B
Sperry Rand Univac	7074	USA	Dec-61	P
IBM	H-400	USA	Dec-61	B
Honeywell	Regnecentralen Gier		Dec-61	P
	Rice Computer aka R1		Dec-61	B
Rice University				

1962

	400 (AN/FSQ 27)	USA	1962	B
TRW	1300	GB	1962	P
ICT	1301	GB	1962	P
ICT	2400/410	USA	1962	P
Philco				

Honeywell	AN/GYK-3(V) (Honeywell D825)	USA	1962	B
	Bailey Meter 756		1962	B
Hughes	HCM-201	USA	1962	I
AMBAC	MICRO computer	USA	1962	I
Packard Bell	PB 440 aka Raytheon 520	USA	1962	G
AEG Telefunken	TR-4	DE	1962	P
NCR	315	USA	Jan-62	E
NCR	315 CRAM	USA	Jan-62	B
Sperry Rand Univac	File II	USA	Jan-62	B
IBM for NSA	HARVEST	USA	Jan-62	A
HRB-Singer	SEMA	USA	Jan-62	B
Elliott	502	GB	Feb-62	P
IBM	1710	USA	Feb-62	B
IBM	1720	USA	Feb-62	B
Olivetti	ELEA 6001	IT	Feb-62	P
Bull	Gamma 30	FR	Feb-62	P
	Linc	USA	Mar-62	B
DEC	LINC	USA	Mar-62	D
NEC	NEAC 2206	JP	Mar-62	P
Oki	OKITAC 5090D	JP	Mar-62	P
Toshiba	TOSBAC 4200	JP	Mar-62	P
ASI	ASI 210	USA	Apr-62	B
LEO	LEO III	GB	Apr-62	P
Telefunken	TR4	DE	Apr-62	D
Hitachi	HITAC 3010	JP	May-62	P
IBM	7072	USA	Jun-62	P
	CE 55		Jun-62	P
Sperry Rand Univac	UNIVAC III	USA	Jun-62	B
GE	412	USA	Jul-62	B
ICT	1500	GB	Jul-62	P
Burroughs	B 260	USA	Jul-62	B
Burroughs	B 270	USA	Jul-62	B
Burroughs	B 280	USA	Jul-62	P
NEC	NEAC 2204	JP	Jul-62	P
DEC	PDP-4	USA	Jul-62	P
U of Illinois	ILLIAC II	USA	Aug-62	A
English Electric	KDP10 (aka RCA 501)	GB	Aug-62	P
SDS	910	USA	Sep-62	P
SDS	920	USA	Sep-62	B
Sperry Rand Univac	1107	USA	Sep-62	P
IBM	7094	USA	Sep-62	P
English Electric	KDN2	GB	Sep-62	P
Collins Radio Co	8400	USA	Nov-62	B
IBM	1620 MOD II (MOD III?)	USA	Dec-62	B
Manchester University	ATLAS	GB	Dec-62	S
Ferranti	Atlas	GB	Dec-62	P
Burroughs	B 2000	USA	Dec-62	B
Datasaab	D21	SE	Dec-62	P
Fujitsu	FACOM 241	JP	Dec-62	P
Hitachi	HITAC 3030	JP	Dec-62	P
CDC	LPG-21	USA	Dec-62	B
Zuse	Z31	DE	Dec-62	P

1963

Sperry Rand Univac	1824	USA	1963	I
IBM	7750	USA	1963	B
Kearfott	AN/ASN-24	USA	1963	I
Marconi	ARCH 1000	GB	1963	A
MIT	Block I AGC	USA	1963	I
	EPSCO 275		1963	P
Bull	Gamma 10	FR	1963	P
IBM NASA	Gemini Guidance Computer	USA	1963	I
Weizmann Inst	Golem	IS	1963	
Hughes	HCM-202	USA	1963	I
IBM NASA	Saturn IB/V LVDC	USA	1963	I
Cambridge University	TITAN	GB	1963	S
	UMC10	PL	1963	B
	ZAM 3	PL	1963	D
ASI	ASI 420	USA	Jan-63	P
Hughes	H330	USA	Jan-63	P
Ferranti	Orion 1	GB	Jan-63	P
H.W.Electronics	15K	USA	Feb-63	B
Burroughs	B 5000	USA	Feb-63	P
NEC	NEAC 2400	JP	Feb-63	P
Philco	TRANSAC S-2000-212	USA	Feb-63	B
Collins Radio Co	Data Central		Mar-63	P
NEC	NEAC 2230	JP	Mar-63	P
Oki	OKITAC 5090H	JP	Mar-63	P
Oki	OKITAC 5090M	JP	Mar-63	P
Elliott	503	GB	Apr-63	P
IBM	7040	USA	Apr-63	B
English Electric	KDF9	GB	Apr-63	P
	PRODAC 500		Apr-63	P
Zuse	Z25	DE	Apr-63	P
Fujitsu	FACOM 231	JP	May-63	P
Westinghouse	510/580	USA	Jun-63	B
Philco	1000	USA	Jun-63	P
CDC	3600	USA	Jun-63	B
CCC	DDP-24	USA	Jun-63	B
Bunker-Ramo	TRW-230	USA	Jun-63	B
IBM	7044	USA	Jul-63	B
Ferranti Packard	FP 6000	CA	Jul-63	P
Bull	Gamma 305	FR	Jul-63	P
GE	215	USA	Sep-63	P
Sperry Rand Univac	1004	USA	Sep-63	B
Sperry Rand Univac	1050 III	USA	Sep-63	P
English Electric	KDF6	GB	Sep-63	P
DEC	PDP-5	USA	Sep-63	P
IBM	1460	USA	Oct-63	E
IBM	7010	USA	Oct-63	P
	CAE 510		Oct-63	P
	CSC-636	AU	Oct-63	B
Datanet	DN-30	USA	Oct-63	P
IBM	1440	USA	Nov-63	E

Honeywell	H-1800	USA	Nov-63	B
Matsushita	MADIC III	JP	Nov-63	P
NEC	NEAC 3800	JP	Nov-63	P
Toshiba	TOSBAC 3300	JP	Nov-63	P
ITT	525 VADE		Dec-63	P
Honeywell	610	USA	Dec-63	P
Honeywell	620	USA	Dec-63	P
Siemens	3003	DE	Dec-63	P
Philco	4000	USA	Dec-63	P
SDS	9300	USA	Dec-63	B
ASI	ASI 2100	USA	Dec-63	B
Honeywell	H-1400	USA	Dec-63	B
Honeywell	H600	USA	Dec-63	P
Librascope	L3055	USA	Dec-63	B

1964

CAS Institute	119	PRC	1964	A
	BESM 6	USSR	1964	I
Adelaide University	CIRRUS	AU	1964	D
SAAB	CK 37	SE	1964	I
Burroughs	D 84	USA	1964	I
College of Saskatoon	M3	CA	1964	A
Matsushita	MADIC 500	JP	1964	P
Mathatronics	Mathatron		1964	E
Mitsubishi	Melcom 1530	JP	1964	P
Dietz	Mincal E	USA	1964	D
Ferranti	Atlas 2	GB	Jan-64	P
Burroughs	B 263	USA	Jan-64	P
Burroughs	B 273	USA	Jan-64	B
Burroughs	B 283	USA	Jan-64	P
	CE 201		Feb-64	P
	DSI 1000		Feb-64	P
Pacific Data Systems	PDS 1020		Feb-64	B
Pacific Data Systems	PDS 1068		Feb-64	P
Raytheon	440	USA	Mar-64	P
IBM	7700	USA	Mar-64	B
CDC	160G	USA	Apr-64	B
GE	235	USA	Apr-64	P
GE	265	USA	Apr-64	P
IBM	7094 II	USA	Apr-64	B
Burroughs	B 160	USA	Apr-64	B
Burroughs	B 170	USA	Apr-64	P
Burroughs	B 180	USA	Apr-64	P
	GE/PAC 4000		Apr-64	P
	GE/PAC 4040		Apr-64	P
GE	415	USA	May-64	B
GE	420	USA	May-64	P
CDC	3200	USA	May-64	B
Hughes	H3118	USA	May-64	P
GEC	90/25	GB	Jun-64	P
GEC	90/30	GB	Jun-64	P
Beckman Instruments	420	USA	Jun-64	B

GE	425	USA	Jun-64	B
SDS	925	USA	Jun-64	P
SDS	930	USA	Jun-64	B
Sperry Rand Univac	1004 II	USA	Jun-64	B
Sperry Rand Univac	1004 III	USA	Jun-64	B
CDC	8092	USA	Jun-64	B
	CAE 90/40		Jun-64	P
	SETI PALLAS		Jun-64	P
Bunker-Ramo	TRW-340	USA	Jun-64	P
Honeywell	200/200	USA	Jul-64	P
GE	205	USA	Jul-64	B
Varian	610	USA	Jul-64	B
RCA	3301	USA	Jul-64	B
CDC	8090	USA	Jul-64	P
	DMI 610		Jul-64	P
Honeywell	H-200	USA	Jul-64	B
NEC	NEAC 2200/200	JP	Jul-64	P
Westinghouse	50	USA	Aug-64	B
Bunker-Ramo	133	USA	Aug-64	P
SEL	820	USA	Aug-64	P
GE	4040	USA	Aug-64	B
	Adage Ambilog-200		Aug-64	B
	PRODAC 50		Aug-64	P
Bull	415	FR	Sep-64	P
Sperry Rand Univac	418	USA	Sep-64	B
Bull	425	FR	Sep-64	P
CDC	6600	USA	Sep-64	R
Ferranti	Orion II	GB	Sep-64	P
AEG Telefunken	TR-10	DE	Sep-64	P
English Electric	KDF8	GB	Oct-64	P
NEC	NEAC 1210	JP	Oct-64	P
DEC	PDP-6	USA	Oct-64	P
NCR	315-100	USA	Nov-64	B
IBM	360/91	USA	Nov-64	E
ICT	1900	GB	Nov-64	P
CDC	3400	USA	Nov-64	B
Burroughs	B 5500	USA	Nov-64	B
Olivetti	ELEA 4001	IT	Nov-64	P
Hitachi	HITAC 4010	JP	Nov-64	P
GEC	90/300	GB	Dec-64	P
ICT	1905E	GB	Dec-64	P
ICT	1905F	GB	Dec-64	P
Digital Electronics	3080	USA	Dec-64	B
	CAE 90/80		Dec-64	P
	DIGIAC 3080		Dec-64	P
DEC	PDP-7	USA	Dec-64	P
Toshiba	TOSBAC 3400	JP	Dec-64	P
Toshiba	TOSBAC 4300	JP	Dec-64	P

1965

CAS Institute	109B	PRC	1965	A
ICT	1904	GB	1965	B

ASI	6080	USA	1965	D
EA	8800	USA	1965	D
Foxboro	97600	USA	1965	D
Foxboro	97600A	USA	1965	D
Bunker-Ramo	BRIC-1	USA	1965	D
Autonetics	D26J	USA	1965	I
CCC	DDP-24A	USA	1965	D
CCC	DDP-24P	USA	1965	D
CCC	DDP-24VM	USA	1965	D
Lear Siegler	DIVAC	USA	1965	I
Beijing Telecomm	DJS-7	PRC	1965	A
Bull	Gamma 135	FR	1965	D
Honeywell	HDC-801	USA	1965	
ITI	ITI 4900	USA	1965	D
Polish Academy of Science	KAR-65	PL	1965	E
English Electric	KDF7	GB	1965	P
Litton	L-304	USA	1965	I
Itek	MCP-1	USA	1965	D
Northrop Corp	NDC-1051	USA	1965	I
Oregon State University	NEBULA	USA	1965	C
DEC	PDP-7A	USA	1965	E
	Serel 505		1965	P
	STC 8300 ADX		1965	P
Toshiba	TOSBAC 3225A	JP	1965	D
Toshiba	TOSBAC 5200	JP	1965	P
ICT	1905	GB	Jan-65	M
GEC	90/2	GB	Feb-65	P
SDS	92	USA	Feb-65	P
CDC	3100	USA	Feb-65	B
	CAE 90/10		Feb-65	P
	L-2010		Feb-65	P
EMR	ADVANCE 6000	USA	Mar-65	P
ASI	ASI 6020	USA	Mar-65	B
CCC	DDP-224	USA	Mar-65	B
Fujitsu	FACOM-230/10	JP	Mar-65	P
Hughes	H3324	USA	Mar-65	P
Hitachi	HITAC 5020	JP	Mar-65	P
Librascope	LGP-21	USA	Mar-65	P
Electrologica	X-8	NL	Mar-65	P
Siemens	303	DE	Apr-65	P
GE	625	USA	Apr-65	B
Elliott	4120	GB	Apr-65	P
CCC	DDP-116	USA	Apr-65	B
LEO	LEO III-326	GB	Apr-65	P
LEO	LEO III-360	GB	Apr-65	P
DEC	PDP-8	USA	Apr-65	B
IBM	360/30	USA	May-65	B
IBM	360/40	GB	May-65	B
GE	635	USA	May-65	P
GE	645	USA	May-65	P
ICT	1904E	GB	May-65	P
ICT	1904F	GB	May-65	P
GE	4060	USA	Jun-65	B

Bull	Gamma M40	FR	Jun-65	P
	GE/PAC 4050 I		Jun-65	P
	GE/PAC 4060		Jun-65	P
Toshiba	TOSBAC 5300	JP	Jun-65	P
Toshiba	TOSBAC 5400/10	JP	Jun-65	P
NCR	315/RMC-501	USA	Jul-65	B
SEL	810-A	USA	Jul-65	P
SEL	840-A	USA	Jul-65	P
ICT	1902	GB	Jul-65	P
ICT	1903	GB	Jul-65	P
EA	8400	USA	Jul-65	P
ASI	ASI 6040	USA	Jul-65	B
Burroughs	B 300	USA	Jul-65	P
	DMI 620		Jul-65	P
ICT	1909	GB	Aug-65	P
IBM	360/50	USA	Sep-65	B
GE	435	USA	Sep-65	P
NCR	500	USA	Sep-65	P
NCR	590	USA	Sep-65	B
Varian	611/612	USA	Sep-65	B
IBM	1130	USA	Sep-65	B
	DMI 611		Sep-65	P
	DMI 612		Sep-65	P
Philips	PR 8000	FR	Sep-65	P
Toshiba	TOSBAC 5400/20	JP	Sep-65	P
Sperry Rand Univac	491	USA	Oct-65	B
Sperry Rand Univac	492	USA	Oct-65	B
Raytheon	520 aka PB-440	USA	Oct-65	B
Elliott	903	GB	Oct-65	P
Siemens	4004/15	DE	Oct-65	P
Siemens	4004/25	DE	Oct-65	P
ASI	ASI 6070	USA	Oct-65	B
ASI	ASI 6240	USA	Oct-65	B
Honeywell	H21	USA	Oct-65	P
Honeywell	H22	USA	Oct-65	P
RCA	Spectra 70/15	USA	Oct-65	B
Philco	102 (CPS)	USA	Nov-65	B
Olivetti	115	IT	Nov-65	P
IBM	360/65	USA	Nov-65	B
Bull	435	FR	Nov-65	P
Varian	620	USA	Nov-65	B
SEL	840	USA	Nov-65	B
Elliott	MCS 920B	GB	Nov-65	P
SCC	SCC 660	USA	Nov-65	P
SCC	SCC 670	USA	Nov-65	P
Honeywell	200/2200	USA	Dec-65	P
Sperry Rand Univac	1108 II	USA	Dec-65	P
CDC	3300	USA	Dec-65	P
CDC	3800	USA	Dec-65	B
Honeywell	H-2200	USA	Dec-65	B
RCA	Spectra 70/25	USA	Dec-65	B

1966

IBM	4Pi/CP	USA	1966	I
IBM	4Pi/EP	USA	1966	I
Sperry Rand Univac	1108	USA	1966	B
Sperry Rand Univac	1830-A	USA	1966	I
TI	2501	USA	1966	I
CDC	5360	USA	1966	I
CDC	5400	USA	1966	I
CDC	6416	USA	1966	B
Computing Devices	AN/UYK-501	CA	1966	I
MIT	Block II AGC	USA	1966	I
Datasaab	D22	SE	1966	S
Digico	Digiac	GB	1966	D
Beijing Telecomm	DJS-6	PRC	1966	A
Bull	Gamma 55	FR	1966	
Hughes	HCM-205	USA	1966	I
Honeywell	HDC-501	USA	1966	I
AC Electronics	MAGIC 321	USA	1966	I
TRW	MARCO 4418	USA	1966	I
Sperry Rand	MARK XII	USA	1966	I
Sperry Rand	MARK XIV	USA	1966	I
Sperry Rand	MARK XVI	USA	1966	I
AMBAC	MICRO D	USA	1966	I
Spear Inc	MicroLinc II	USA	1966	D
	MINSK-222	USSR	1966	P
GECMarconi	Myriad II	GB	1966	C
Northrop Corp	NDC-1051A	USA	1966	I
Electrologica	X-5	NL	1966	P
Honeywell	200/1200	USA	Jan-66	P
IBM	360/75	USA	Jan-66	B
Friden	6010	USA	Jan-66	B
CCC	DDP-124	USA	Jan-66	B
Honeywell	H-1200	USA	Jan-66	B
Hughes	H3118 M	USA	Jan-66	P
Zuse	Z32	DE	Jan-66	P
Honeywell	200/120	USA	Feb-66	P
Sperry Rand Univac	1005 I	USA	Feb-66	B
Sperry Rand Univac	1005 II	USA	Feb-66	B
Sperry Rand Univac	1005 III	USA	Feb-66	B
IBM	1800	USA	Feb-66	B
ASI	ASI 6050	USA	Feb-66	B
Honeywell	H-21	USA	Feb-66	B
Honeywell	H-120	USA	Feb-66	B
IBM	360/20	USA	Mar-1966	B
IBM	360/67	USA	Mar-66	P
Sperry Rand Univac	494	USA	Mar-66	P
CDC	1700	USA	Mar-66	P
Fujitsu	FACOM-230/50	JP	Mar-66	P
Bull	Gamma 115	FR	Mar-66	P
Hughes	HM4118	USA	Mar-66	P
GE	115	USA	Apr-66	P
SDS	940	USA	Apr-66	B
CDC	6400	USA	Apr-66	P

SCC	SCC 650	USA	Apr-66	P
	SEA 4000		May-66	P
	CCD 516	DDR	Jun-66	B
Clary Computer	DE 600	USA	Jun-66	C
	GE/PAC 4050 II		Jun-66	P
SCC	SCC 665	USA	Jun-66	P
Electrologica	X-2	NL	Jun-66	P
Electrologica	X-4	NL	Jun-66	P
Siemens	4004/45	DE	Jul-66	P
Elliott	4130	GB	Jul-66	P
DEC	Linc-8	USA	Jul-66	P
RCA	Spectra 70/45	USA	Jul-66	B
RCA	Spectra 70/55	USA	Jul-66	B
Electrologica	X-3	NL	Jul-66	P
DEC	PDP-9	USA	Aug-66	P
SDS	Sigma 2	USA	Aug-66	C
ICT	1901	GB	Sep-66	P
Fujitsu	FACOM-230/20	JP	Sep-66	P
Hitachi	HITAC 8200	JP	Sep-66	P
Mitsubishi	Melcom 3100/10	JP	Sep-66	P
Mitsubishi	Melcom 3100/30	JP	Sep-66	P
Mitsubishi	Melcom 3100/50	JP	Sep-66	P
DEC	PDP-8/S	USA	Sep-66	P
IBM	360/44	USA	Oct-66	B
Philco	2000-213	USA	Oct-66	G
	CAE 10070		Oct-66	P
CCC	DDP-516	USA	Oct-66	P
	GE/PAC 4020		Oct-66	P
NEC	NEAC 2200/400	JP	Oct-66	P
GEC	S.7	GB	Oct-66	P
RCA	Spectra 70/35	USA	Oct-66	P
HP	2116-A	USA	Nov-66	B
Data Mate	ECP-18		Nov-66	B
	IC 6000/19		Nov-66	P
	IC 6000/29		Nov-66	P
	IC 6000/39		Nov-66	P
NEC	NEAC 2200/100	JP	Nov-66	P
NEC	NEAC 2200/500	JP	Nov-66	P
BIT Inc	80	USA	Dec-66	B
BIT Inc	480	USA	Dec-66	P
ICT	1906E	GB	Dec-66	P
ICT	1906F	GB	Dec-66	P
ICL	1907E	GB	Dec-66	P
ICT	1907F	GB	Dec-66	P
Siemens	4004/55	DE	Dec-66	P
GE	4050	USA	Dec-66	B
Hitachi	HITAC 5020E	JP	Dec-66	P
Hitachi	HITAC 8100	JP	Dec-66	P
	SEA 1500		Dec-66	P
SDS	Sigma 7	USA	Dec-66	B
Toshiba	TOSBAC 5100/20	JP	Dec-66	P

1967

IBM	4Pi/TC	USA	1967	I
CAS Institute	109C	PRC	1967	A
CDC	449	USA	1967	I
Elliott	920M	GB	1967	I
Sperry Rand Univac	1818	USA	1967	I
ICT	1906	GB	1967	M
ICT	1907	GB	1967	M
CDC	5400 B	USA	1967	I
CDC	6500	USA	1967	B
Collins Radio Co	C-8311A-1	USA	1967	I
Kearfott	GPK-10	USA	1967	I
Kearfott	GPK-20	USA	1967	I
	M-1000	USSR	1967	C
	M-2000	USSR	1967	C
	M-3000	USSR	1967	C
AC Electronics	MAGIC 301	USA	1967	I
AC Electronics	MAGIC 311	USA	1967	I
AC Electronics	MAGIC 331	USA	1967	I
Northrop Corp	NDC-1060	USA	1967	I
Raytheon	RAC-230	USA	1967	I
Raytheon	RAC-250	USA	1967	I
Regnecentralen	RC 4000	DK	1967	
Teledyne	Series 20000	USA	1967	I
RCA	VIC-36A	USA	1967	I
EMR	ADVANCE 6130	USA	Jan-67	P
Burroughs	B 8500	USA	Jan-67	P
	C-8500		Jan-67	P
Hitachi	HITAC 8300	JP	Jan-67	P
English Electric	System 4/10	GB	Jan-67	P
IBM	360/90	USA	Feb-67	P
EA	640	USA	Feb-67	P
Siemens	4004/35	DE	Feb-67	P
GE	4020	USA	Feb-67	B
NEC	NEAC 1240	JP	Feb-67	P
NEC	NEAC 2200/300	JP	Feb-67	P
GEC	S.2	GB	Feb-67	P
CDC	3500	USA	Mar-67	P
Bull	Gamma 140	FR	Mar-67	P
English Electric	System 4/30	GB	Mar-67	P
CCC	DDP-416	USA	Apr-67	P
Bull	Gamma 145	FR	Apr-67	P
Interdata	3	USA	May-67	B
NCR	315/RMC-502	USA	May-67	P
Burroughs	B 2500	USA	May-67	P
Burroughs	B 3500	USA	May-67	B
Honeywell	H-4200	USA	May-67	B
SCC	IC-6000	USA	May-67	B
NEC	NEAC 2200/50	JP	May-67	P
NEC	NEAC 3100	JP	May-67	P
Zuse	Z26	DE	May-67	P
Sperry Rand Univac	920	USA	Jun-67	P
IBM	1401-H	USA	Jun-67	B

CDC	6800	USA	Jun-67	P
Sperry Rand Univac	9200	USA	Jun-67	B
Toshiba	TOSBAC 7000/60	JP	Jun-67	P
AEG Telefunken	TR-86	DE	Jun-67	P
AEG Telefunken	TR-440	DE	Jun-67	P
SEL	840-MP	USA	Jul-67	P
Xerox	Sigma 5	USA	Aug-67	B
Westinghouse	250 (S-2)	USA	Sep-67	B
Siemens	302	DE	Sep-67	P
Raytheon	703	USA	Sep-67	P
Sperry Rand Univac	9300	USA	Sep-67	P
DEC	PDP-10 (KA10)	USA	Sep-67	P
	PRODAC 250		Sep-67	P
SCC	SCC 6700	USA	Sep-67	P
English Electric	System 4/50	GB	Sep-67	P
Toshiba	TOSBAC 5100/30	JP	Sep-67	P
Digital Electronics	3080-C	USA	Oct-67	B
ASI	ASI 6130	USA	Oct-67	B
Siemens	305	DE	Nov-67	P
HP	2115-A	USA	Nov-67	I
CII	10020	FR	Nov-67	P
Hitachi	HITAC 8400	JP	Nov-67	P
Honeywell	125	USA	Dec-67	B
Hitachi	HITAC 8500	JP	Dec-67	P
English Electric	System 4/70	GB	Dec-67	P

1968

ICT	1901A	GB	1968	B
ICT	1902A	GB	1968	B
ICT	1903A	GB	1968	B
ICT	1904A	GB	1968	B
ICT	1906A	GB	1968	W
TI	2540	USA	1968	I
TI	2550	USA	1968	I
CDC	5100	USA	1968	I
Shanghai Institute	C2	PRC	1968	A
CII	IRIS 50	FR	1968	
Digico	Micro 16	GB	1968	C
	Mobidic		1968	B
Computer Technology	Modular 1	GB	1968	J
Northrop Corp	NDC-1070	USA	1968	I
Westinghouse	OBP	USA	1968	I
AMBAC	portable MICRO D	USA	1968	I
CNES	Roseau Computer	FR	1968	I
CII	10010	FR	Jan-68	P
Burroughs	B 6500	USA	Jan-68	P
Burroughs	B 7500	USA	Jan-68	P
General Automation	SPC-8	USA	Jan-68	B
Honeywell	200/4200	USA	Feb-68	P
GE	405	USA	Feb-68	P
Mitsubishi	Melcom 9100/30	JP	Mar-68	P
Interdata	2	USA	Apr-68	P

Interdata	4	USA	Apr-68	P
DEC	PDP-8/I	USA	Apr-68	B
Computer Automation	PDC 808	USA	May-68	B
Honeywell	200/8200	USA	Jun-68	P
Siemens	304	DE	Jun-68	P
Honeywell	1648	USA	Jun-68	B
Motorola	MDP-1000	USA	Jun-68	D
Redcor Corporation	RC-70	USA	Jul-68	B
Honeywell	1250	USA	Aug-68	B
AEG Telefunken	TR-84	DE	Aug-68	P
Honeywell	110	USA	Sep-68	B
Varian	520/i	USA	Sep-68	D
HP	2116-B	USA	Sep-68	I
HP	9100A	USA	Sep-68	S
HP	9100B	USA	Sep-68	S
NCR	Century 100	USA	Sep-68	B
SCC	IC-4000	USA	Sep-68	B
IBM	360/25	USA	Oct-68	B
HP	2114-A	USA	Oct-68	B
HP	2000-A	USA	Nov-68	B
Burroughs	B 500	USA	Nov-68	B
DEC	PDP-8/L	USA	Nov-68	B
DEC	PDP-9/L	USA	Nov-68	B
RCA	Spectra 70/46	USA	Nov-68	B
Honeywell	632	USA	Dec-68	B
SEL	810-B	USA	Dec-68	B
	Bailey Meter 855		Dec-68	B
Honeywell	CCD 416	USA	Dec-68	B
Dynamic Research	DRC-44	USA	Dec-68	B
Computer Automation	PDC 816	USA	Dec-68	B

1969

RCA	200 series	USA	1969	I
Siemens	301	DE	1969	D
AMBAC	1808	USA	1969	I
Sperry Rand Univac	1819	USA	1969	I
ICL	1908A	GB	1969	B
CDC	Alpha	USA	1969	I
COMCET	Comcet 10	USA	1969	D
COMCET	Comcet 40	USA	1969	D
COMCET	Comcet 60	USA	1969	D
Datasaab	D5	SE	1969	S
Autonetics	D200-1	USA	1969	I
	EPOS-2	CS	1969	C
Honeywell	HDC-201	USA	1969	I
SCC	IC-9000	USA	1969	S
BBN	IMP	USA	1969	
Litton	L-3050	USA	1969	I
Lear Siegler	LS-50	USA	1969	I
Digital Scientific	META 4	USA	1969	C
	MPP	USA	1969	I
Qantel	Qantel V	USA	1969	D

General Automation	SPC-12	USA	1969	I
Westinghouse	WADSP 1802	USA	1969	I
Wilkinson	WCS-881	USA	1969	D
Interdata	15	USA	Jan-69	B
Micro Systems	800	USA	Jan-69	D
Micro Systems	810	USA	Jan-69	D
CDC	7600	USA	Jan-69	B
Honeywell	8200	USA	Feb-69	B
Data General	Nova	USA	Feb-69	B
GE	105	USA	Mar-69	B
Raytheon	704	USA	Mar-69	B
Lockheed Electronics	MAC-16	USA	Mar-69	D
	Sanders 200		Mar-69	B
GE	30	USA	Apr-69	B
DEC	PDP-12	USA	Apr-69	B
SCC	SCC 4700	USA	Apr-69	B
Varian	620/i	USA	May-69	D
Raytheon	706	USA	May-69	B
Data Craft	6024-1		May-69	B
Sperry Rand Univac	9400	USA	May-69	B
SCC	DCT-132	USA	May-69	B
GE	615	USA	Jun-69	B
NCR	Century 200	USA	Jun-69	B
SCC	IC-7000	USA	Jun-69	B
Computer Automation	208	USA	Jul-69	D
Honeywell	CCD 316	USA	Jul-69	B
Xerox	CE-16	USA	Jul-69	B
General Automation	GA-18/20	USA	Jul-69	B
Sprias Systems	65	NL	Aug-69	B
Westinghouse	P-2000	USA	Aug-69	D
Tempo Computers	1/1-A	USA	Sep-69	D
Xerox	CF-16	USA	Sep-69	B
CII	IRIS 80	FR	Sep-69	
IBM	360/85	USA	Oct-69	B
CDC	6700	USA	Oct-69	B
Data Mate	DM-16	USA	Oct-69	B
GE	58	USA	Nov-69	B
HP	2114-B	USA	Nov-69	I
Computer Logic	CLS-18	USA	Nov-69	B
SCC	DCT-32	USA	Nov-69	B
BIT Inc	483	USA	Dec-69	D
Micro Systems	812	USA	Dec-69	B
Sperry Rand Univac	1106	USA	Dec-69	B
Xerox	Sigma 3	USA	Dec-69	D

Sayonara, Y2K

The time has come for me to say goodbye to Y2K. I'm removing the Y2K articles, links, and resources from my web site, and dropping off the Y2K radar screen. Y2K has been part of my life since early 1995 and has occupied nearly every waking moment since the summer of 1997 -- and while it will continue to have a significant impact on my personal and family life, I no longer plan to play a public role. I realize that this may raise some questions, and perhaps cause some commentary and debate, and I've attempted to answer the more obvious questions below.

Why? Why Now?

No, I haven't been abducted by aliens. I haven't been kidnapped by the CIA or the mysterious people in black helicopters. I haven't been threatened by the FBI. I haven't been bribed by banks or the government (or anyone else). There is nothing deep, dark, or mysterious about my decision.

I simply feel that I've done everything I can do to raise the alarm about Y2K. I've co-authored two books, written dozens of articles and essays, spoken at hundreds of seminars, conferences, meetings, and gatherings. I could continue doing the same thing, over and over again, but I would be repeating myself. More important, I would be preaching to the choir; those whose opinion and outlook on Y2K are compatible with mine would nod their head in agreement, and those whose opinion and outlook are incompatible with mine would shake their head in disbelief, just as they have for the past four years.

Yes, there are still some people who are undecided, and who continue to listen to both the optimists and the pessimists before making up their own mind. But I think that a more accurate term for "undecided" is "indifferent" -- i.e., there are many people who still don't care, who don't think the topic is worthy of serious attention, and who may not focus on Y2K until this fall -- and quite possibly not even until midnight on New Year's Eve. Meanwhile, I sense a hardening of positions: those who are pessmistic about the outcome are even more convinced than they were a year ago, and those who are optimistic are even more convinced, especially because they see a steady stream of upbeat press releases and government status reports.

More than just hardening of positions, though, I sense an increasing degree of confrontation and hostility between the two camps. It's reflected in flame wars on the Internet discussion groups; emotional rhetoric in the statements of government officials and media articles (e.g., warnings against "frivolous stockpiling"); McCarthy-esque threats by both sides that "we're taking names" in preparation for some kind of undescribed post-Y2K retribution against those who express an opposing point of view; and, overall, a sharp decline in civility. I expect this to continue for the remainder of the year, and I don't think it's a productive use of my time (or anyone else's) to continue attempting to respond to messages and commentary whose purpose often seems to be "ignore the message, shoot the messenger." Why isn't it productive? Because it doesn't change anyone's mind about the topic. Perhaps we could use the services of some of the gifted statesmen who have helped negotiate peace treaties in northern Ireland, or the middle East; as for me, I don't have the skill, the patience, or the training in this kind of diplomacy.

Does This Imply A Change of Opinion About Y2K?

No doubt there will be some who gleefully proclaim, "This just proves that Ed was wrong about Y2K all along! He has given up on his 'doomer' position, but he's too much of a coward to say so!" Well, time will tell whether any of us were right or wrong about Y2K -- but for now, my perspective on

Y2K remains essentially unchanged. I stand by the comments I've made in all of the articles and essays that I've written; at a "macro" level, I still have a pessimistic outlook about the outcome of Y2K. We can argue indefinitely about whether the large government agencies and the large companies in England, Canada, Australia, and the U.S. will manage to muddle through, and whether the failure or bankruptcy of a few such organizations and/or agencies will have a dramatic impact. But even amongst the optimists, there seems to be a common consensus that small businesses, small towns/counties, and small (aka Third World) countries are so far behind that they're unlikely to finish repairing a signficant percentage of their mission-critical systems. The "fortress America" attitude amongst the optimists seems to be, "Well, so what if half of the small businesses don't do anything about Y2K until they see what breaks? So what if Eastern Europe, Africa, South America, the Middle East, and most of Asia don't manage to repair their systems? Why should I believe that this will have any impact on *my* life?"

Similarly, we can argue indefinitely about whether the governmental authorities and the private-sector organizations (e.g., the banks, the utilities, the telephone companies, etc.) are doing a good job or a bad job in terms of reporting their status and progress to the public. But there is a widespread theme that they're in control of the situation, and that (notwithstanding the possibility of a few "glitches") there's really nothing serious to worry about. Yet the federal government has acknowledged that it doesn't have the resources to provide emergency relief to all of the local towns and communities across the country; local communities are being told that they're on their own, and that they should make their own contingency plans. But the federal government can't force them to do so, nor can it force small businesses to make Y2K a top priority, nor can it issue ultimatums to foreign governments to do anything about Y2K. I don't even think it has control over the outcome of Y2K repairs within its own agencies, for non-compliant code doesn't listen to the rhetoric of politicians -- it either works, or it doesn't work. Ultimately, there is only a limited amount of control that corporations and government agencies have over the technological outcome of Y2K; yet the prevailing attitude seems to be that government and industry *are* in control, as long as they can "manage" the perceptions of the public. I have believed, all along, that Y2K is too big, too complex, and too systemic in nature to be "controlled" from a technological perspective; and I believe that the public's *perception* of Y2K will ultimately be shaped by tangible events that impact their lives, much more than it's shaped by the "spin control" efforts of government and industry. For the past few months, the PR spin control has been quite effective, and I fully expect that it will continue throughout the summer as government and industry seek to "reassure" the public. And since the public would generally prefer to be reassured that the government is taking care of any problems looming on the horizon, rather than face the possibility of serious disruptions, the spin control efforts may continue succeeding even into the fall of 1999.

Those who want me to continue participating in the public debate sometimes ask me, "But isn't it possible that things will change in the final months of Y2K?" And the optimists ask a roughly similar question: "Yes, I agree that things look bad in small companies, small towns, and small countries -- and maybe even in some of the big companies and big agencies. But don't you agree that with a lot of hard work, we can redouble our efforts, achieve a quantum leap in productivity, and make enough progress in these last few months to avert disaster?" To which my answer is, quite simply, "No." If you believe in the Tooth Fairy, or in the kind of implausible miracles favored by Hollywood script-writers, then perhaps you can sustain your belief that everything will somehow work out in the end. If you're looking at an individual company, or an individual government agency, perhaps you can make a plausible case -- yes, sometimes we get lucky, sometimes the combination of inspiration and perspiration are sufficient to overcome enormous odds. But at the macro level, I don't think it makes sense. We have 30 years of data in the software field that tells what to expect in the "average" case -- i.e., 25% of all projects are cancelled, 15% are delivered behind schedule, and the resulting systems have an average of one defect for every thousand lines of code.

If a miracle were to occur, it would have occurred two, or three, or four years ago. If President Clinton had addressed a joint session of Congress in 1996 and declared a state of emergency until Y2K had been completely conquered, perhaps we could look forward to a successful outcome at the end of this year. I'm not talking about the martial-law, conspiracy-theory form of "state of

emergency," but rather a "fireside chat," followed by a series of actions that would make Y2K the highest-priority activity in the land. It didn't happen then, and it isn't happening now. I'm fairly convinced that it won't happen during the remaining seven months of 1999 -- and even if it did, it's now too late. If a high-level executive issues a thundering edict to the Y2K programming staff, "Redouble your efforts! Work harder!", the response from the programmers is likely to be, "Boss, we're thinking as hard as we can!" Software is an intellectual activity, rather than something requiring brawn and muscle-power; you simply can't order people to think harder.

I believe that we are entering the "end game" of Y2K, and that the outcome isn't likely to be changed significantly because of last-minute strategies, edicts, proclamations, or demands for deathmarch-style overtime on the part of programmers. About the only thing that's still an option, both for organizations and for individuals, is contingency planning and preparations for some degree of disruptions. But again, this involves preaching to the choir: those who believe it makes sense to develop *and implement* contingency plans, are already doing so -- indeed, some 90% of private-sector organizations are planning "war rooms" or "control centers" to cope with whatever problems arise. Meanwhile, those who think it's unnecessary will continue to do nothing. Yes, it's possible that there will be a last-minute surge in preparedness activities, especially at the personal level; but it probably won't happen until this fall, at which point it will lead to the very phenomenon of shortages and panics that government spokesmen have been warning about. Meanwhile, it's going to be a long, hot, quiet summer of Y2K-denial, unless some significant, undeniable, *tangible* event occurs.

What About All The People Who Don't Know About Y2K?

When I decided to move from New York City to New Mexico last year, some of the Y2K activists criticized me severely for "abandoning" New York. "You've doomed eight million innocent citizens to their fate!" I was told. "It's your responsibility to stay in New York, and warn all those people -- so they'll be ready for Y2K!" What a mind-boggling concept! If 8 million oblivious residents of New York City are entirely dependent on me, or any other individual, to learn what Y2K is all about, then we're all in a lot more trouble than we ever imagined.

There is no shortage of information about Y2K. If the 8 million New Yorkers, or the 250 million Americans, or the 5 billion citizens of the world, want to know all about Y2K, there are dozens of books, thousands of articles, and tens of thousands of references on the Internet. Ignorance was a plausible excuse in 1995 and 1996, perhaps even in 1997 -- but not now. If someone doesn't know about Y2K, it's because they've chosen to ignore it, and/or because they believe the assurances of government and industry spokesmen who tell them there is nothing to worry about.

In terms of personal responsibility, I am my brother's keeper. Actually, I don't have a brother, but I do have five sisters for whom I feel a sense of responsibility, along with my children, my wife, and my parents. I also feel some degree of responsibility for my neighbors and my community -- partly because I have a personal relationship with many of them, and also because it will do little good for for my family to be personally prepared if my neighbors are not. Beyond that -- i.e., at the state, national, or global level -- I've been happy to spend a considerable amount of my time and effort helping those who are helping themselves. And because I've enjoyed a good living in the computer field that was at least partially responsible for having created and perpetuated the Y2K problem, I've felt a professional responsibility to ensure that people understand what the problem is all about, and why it has been so difficult to solve. But there comes a time when it seems appropriate to say, "Okay, I've done my best to tell you what's going on. Now it's up to you to decide what (if anything) you're going to do about it." For me, that time has come.

Conclusion

I suspect that there are also a number of Y2K activists who will be frustrated that they can no longer send me email messages, asking meto provide an interpretation or analysis of the day-to-day Y2K

announcements from the media and the corporate PR departments. To which I offer two responses: (1) you're intelligent adults, and you can use your own common sense to decide how to interpret the news; and (2) the debate between the optimists and pessimists will continue, with ever more emotion and rhetoric, right up to Jan 1, 2000 and beyond. If you're waiting for someone to produce an absolute, guaranteed, indisputable "answer" to the Y2K debate, you've already waited too long. It's not going to happen. As I suggested in one of my earlier essays, everyone will have to decide for themselves when the "moment of truth" has arrived, when they will make a decision about their own personal Y2K plans, in the presence of incomplete, fuzzy information.

I also suspect that there are a number of Y2K activists who will continue doing everything in their power to raise the alarm, alert the government, and encourage their neighbors and fellow citizens to stockpile and prepare -- right up to the last moment. They have my respect, my admiration, and my best wishes. As for me, it's time to get back to providing for my family.

If there are any major developments this summer or fall, where I think my background and experience in software engineering might provide a useful perspective, I'll dust off my soap-box and offer an appropriate commentary. And when the dust settles, in the days and weeks after Jan 1, 2000, I'll reappear to offer an appropriate *mea culpa* if my Y2K outlook proved wrong.

Meanwhile, my best wishes for everyone as we move into the Y2K end game. It's time for me to say, "Sayonara, Y2K." I'll see you on the other side.

My Y2K Outlook: A Year of Disruptions, a Decade of Depression

Most of us who work in the Y2K field have an instinctive answer when asked, "How bad do you think it will be?" For various reasons, we almost always say, "*I don't know*." That's been my answer for nearly four years, and even today, as we move further into 1999, I have to repeat: *I don't know*. For a more elaborate version of this answer, you might want to take a look at another of my essays, entitled The Y2K Crystal Ball: What's Going to Happen on 1 Jan 2000?

But sooner or later, we all have to "bite the bullet" and make some firm decisions about Y2K, based on our best guess and whatever information we've been able to collect. I'm sure there will be some people who abdicate such a firm decision, because it can be pretty scary: if you really think Y2K is going to cause major disruptions, it requires some expensive, unpleasant, unpopular, and potentially risky actions on your part. Thus, many people will continue singing the Y2K folk song, "I don't know," right up to the stroke of midnight on December 31, 1999. Some are convinced that they can wait until the summer or fall, by which time they hope to have a more definitive picture of the Y2K situation, before committing themselves to a course of action.

For some of us, though, the moment of truth has already arrived. For some of us, there is an epiphany, a light bulb that goes off in our mind, in which we say, "Even if I don't *know* how bad Y2K is going to be, I have to make some plans." Almost all of us have family members we love; many of us have both young children and older parents, for whom we feel responsible. Most of us have assets to protect, though young people just entering the work force may be less concerned about this issue. But those of us in our 30s, 40s, 50s, and 60s may have a house that represents the majority of our net worth; we may have retirement funds, a stock-market account, or perhaps just a couple thousand dollars in a savings account. Not making a decision about all of this is, in itself, a decision: it means that we've decided to subject ourselves to the fickle finger of Fate. If we're lucky, Y2K will turn out to be only a minor bump in the road, and we'll sigh a huge breath of relief. If we're not lucky ... well, those who abdicate making a decision would rather not think about it, or would prefer to amass as many arguments as possible to "prove" that Y2K won't cause major disruptions.

I've been pretty careful to avoid making any specific predictions about Y2K, because I, too, *don't know*. And I'm concerned about the people who send me e-mail, or who read our *Time Bomb 2000* book, who often seem intellectually lazy: rather than doing the homework to make an informed decision, they want someone else to tell them. In a world of information overload, this is understandable: we depend on restaurant critics to tell us whether that new French restaurant is any good, we depend on movie critics to tell us whether that latest action thriller is too bloody, and we even seem to depend on the "talking heads" on television talk-shows to make up our minds for us about the outcome of the President's impeachment hearings. Analysts and commentators serve a useful role for such purposes, but when it comes to really serious issues, it seems to me that every responsible adult should do enough homework to make up his or her mind; thus, I wanted to avoid creating a situation where someone might complain, somewhere down the road, "Well, I made all of my decisions based on Ed Yourdon's recommendations, and he turned out to be totally wrong -- and as far as I'm concerned, all the problems I'm having now are entirely his fault!"

Nevertheless, it turns out that people *have* been making Y2K decisions based on my comments,

books, articles, speeches, and essays. A friend with 30+ years of experience in the computer field recently wrote to me to say that our *Time Bomb 2000* book had been the catalyst in his family's decision to retire from the computer field and move to Belize. Others have made somewhat less radical decisions, but it's clear that at least a small percentage of thoughtful people across the country (if not around the world) are changing jobs, re-distributing their assets, stockpiling necessities, and occasionally moving from what they perceive to be a high-risk geographical location to someplace they hope will be safer. Meanwhile, a much larger majority has internalized all of the information they've seen and heard about Y2K, and have decided to either (a) ignore it, or (b) postpone their decision until sometime in the summer or fall of 1999. For those in category (a), I wish you well and hope that you turn out to be right; for those in category (b), I fear that your options may have vanished by the time you decide to take action, in which case my opinions on Y2K won't matter very much to you.

Thus, I don't think I'll be exerting an inappropriate influence on most of the population if I publish my opinion on the likely consequences of Y2K. After all, I'm just one of many voices in the Y2K field: Peter de Jager has published his opinion in *Scientific American*, the Gartner Group has published its opinion in a number of recent press reports, the President's Y2K Conversion Council has offered its assessment that the consequences of Y2K will probably be no worse than a winter snowstorm. My opinion, as you'll see below, is substantially different.

How Bad Will Y2K Be?

As the title of this essay has already informed you, my opinion is that we're going to suffer a year of technological disruptions, followed by a decade of depression. It may turn out to be less severe, it may turn out to be more severe; it may turn out that I'm totally wrong in my conclusions. But this is *my* assessment, and it forms the basis of my own plans for coping with Y2K; if it turns out that I'm wrong, I'll be spending a lot more time apologizing to my family members than to everyone else who should have been evaluating the same data that I've been evaluating, in order to reach their own opinions.

What Do I Mean By A Year of Disruptions And A Decade of Depression?

Words like "disruption" and "depression" are rather fuzzy, and they mean different things to different people. What I mean by "disruption" is a problem that's large enough and serious enough that it forces me to change my plans and behavior, if not my overall lifestyle; most of all, it means that the impact will be noticeable. Let me illustrate with an example: driving from my home in northern New Mexico to the airport in Albuquerque takes approximately 2.5 hours under normal conditions, but I generally add an extra half-hour for minor delays and traffic jams. That's my "normal" plan, and I would never bother complaining that it took an extra ten minutes to navigate through the traffic in Espanola, or that there was a minor traffic accident in Santa Fe. On the other hand, if the highway was washed out by a flash flood, and if that unanticipated problem added an extra two hours to my trip, there's a good chance that I would miss my flight -- which would probably have consequences ranging from minor to moderate to potentially serious. *That* is a disruption.

The reason I mention this is that recent surveys conducted by Professor Howard Rubin for the Cap Gemini consulting firm indicate that nearly 50% of U.S. companies have *already* experienced some Y2K problems through the end of 1998; even more interesting, nearly 98% expect to experience problems during 1999. *But that doesn't necessarily mean that consumers and citizens will be aware of the problems, or that it will cause a noticeable disruption in their day-to-day lives.* Chances are

that the average consumer experienced dozens, if not hundreds, of minor annoyances and problems in his or her day-to-day life last year -- and while some of those problems might have been caused by Y2K computer glitches, they weren't recognized as such. The classic example is the credit card renewal problem that existed throughout 1996, 1997, and even part of 1998: it didn't occur to us that the credit-card companies were issuing renewal cards with a 1999 expiration date because merchants weren't able to process a Y2K expiration date. It's a minor inconvenience, in the sense that we have to be prepared to deal with the *next* renewal card earlier than would otherwise have been the case -- but it's hardly a major disruption in our lives.

But I believe that we *will* begin seeing Y2K problems that *do* cause noticeable disruptions in our day to day lives; I believe we'll start seeing them by this summer, and I believe they'll continue for at least a year. As many people are now aware, 46 states (along with Australia and New Zealand) will begin their 1999-2000 fiscal year on July 1, 1999; New York (and Canada) will already have gone through their Y2K fiscal rollover on April 1, and the remaining three states begin their new fiscal year on August 1, September 1, and October 1. We also have the GPS rollover problem to look forward to on August 22nd, as well as the Federal government's new fiscal year on October 1st. There is, of course, some finite probability that all of these rollover events will occur without any problems; but there's also a finite probability that pigs will learn to fly.

It's more likely, in my opinion, that problems will emerge in tax systems, welfare systems, pension systems, unemployment systems, food-stamp systems, and various other financial systems operated by the state authorities; and it's also quite possible that these problems will "trickle down" to the county and city level. None of these problems will cause civilization to come to a screeching halt, but they will certainly be regarded as a disruption by those who are expecting pensions checks, food stamps, welfare checks, or other financial interactions with the state government. Of course, the average citizen might say, "Who cares? I'm not unemployed, and I don't get food stamps. I'm not retired, so it doesn't matter to me if the pension checks are delayed." But a disruption doesn't have to be something that affects you directly; if it has a significant impact on your children, or your retired parents, or a sibling, or a close friend, then it's still a disruption. If your brother-in-law calls you up to borrow the money to pay the rent, because his pension check didn't arrive on time, you will definitely feel that you've been affected by the Y2K bug.

Notice that these examples don't involve embedded systems -- and because of that, they probably won't involve life-and-death situations. The embedded system failures are likely to begin surfacing, in a fashion noticeable to the average consumer, about a month before January 1, 2000 (because some of these embedded systems have a "look-ahead" calculation that involves dates a month into the future); they'll probably peak on or about January 1st; and then they'll gradually subside over the next few months. And lots of other business applications and financial systems can also be expected to encounter a larger number of problems on or about January 1st; for reasons that I'll describe in more detail below, I believe they'll continue for at least six months into calendar year 2000.

None of this necessarily means that the electricity will be out for six months, let alone a full year; none of it means that we should necessarily expect to lose nationwide telephone service for a full year; none of this means necessarily that we should expect our bank to be closed for six months. But it *does* mean that we're likely to be living in an environment much like the Third World countries some of us have visited, where nothing works particularly well. One day the phones will be out; the next day, the phones will work but the air-conditioning in our office building will be down; that will be fixed a couple days later, but then the banks will decide to close for a day because of some unexplained problem. North American and European business executives who try to open a regional

office in this kind of environment can often be heard to mutter, "It's just one damn thing after another. You never know, from one day to the next, what's going to work and what's not." And in that kind of environment, it will be difficult (if not impossible) to maintain the level of productivity and efficiency that many of us are accustomed to; we may be forced to adjust to this by accepting the *manana* attitude that citizens in many developing nations take for granted. If we don't get it done today (because of Y2K disruptions), we'll get it done tomorrow .... or next week ... or next month ... or maybe we'll decide that it wasn't so important after all, and we simply won't do it.

Let me put this into somewhat more personal terms. My personal expectation is that my day-to-day work will suffer an increasing number of interruptions, glitches, delays, inconveniences, and disruptions during the second half of 1999; and I'm expecting that they'll be sufficiently pervasive after January 1, 2000 that my income will drop to zero during the first six months of the new year. I work as a computer consultant, which requires that the infrastructure (phones, electricity, etc.) be reasonably dependable; it also means that I have to get on an airplane to fly to my clients' offices, since none of my work is based in the town where I live. If the phone lines are down intermittently throughout the day, it will be much more difficult to get any work done on the Internet. If commercial air traffic is disrupted, I'm grounded without income. If my clients are disrupted by their own problems, chances are they'll cancel the kind of advanced consulting services that I provide.

If I'm the only person on earth to suffer such problems, it obviously won't have a noticeable impact on the global economy. But I believe that *lots* of companies, as well as individuals, are going to be coping with similar problems. Contrary to what many economists are still predicting, I don't think Y2K will be a one-time "bump in the road" that shaves 0.3% off the GNP during the first quarter of 2000, to be followed by a resumption of a booming economy. I believe that Y2K will be equivalent to throwing a million monkey wrenches into the "engine" of the global economy, and that it will lead to a depression similar in severity and duration to the Great Depression. Obviously, it can be rather discouraging to have such a picture of the future in your mind; but it's not the end of the world. By analogy, if someone had been firmly convinced in the spring of 1929 that a crash was inevitable, and that it would be followed by a decade of economic depression, he/she could have made appropriate plans to minimize the personal impact on his/her life. I plan to do the same with Y2K.

What's The Basis For My Opinion?

I can't prove that my view of the future is correct. But my predictions of technological disruptions are based on several things:

- Large companies won't finish their Y2K projects; even if they manage to finish their mission-critical projects, they'll be spending most of 2000 fixing all of the "non-mission-critical" systems they didn't get around to. Yes, of course, the big companies are working hard on the problem; and yes, they're busily issuing optimistic press releases about their progress. I fully expect that 80% of the large U.S. companies and government agencies will get 80% of their mission-critical systems finished; that degree of success is encouraging, and almost certainly represents an argument against TEOTWAWKI. But what about the other 20%? What happens if 20% of the *Fortune* 500 companies fail miserably in their Y2K efforts? What happens if 20% of the 24 major Federal agencies don't finish most of their mission-critical systems in time? What happens if 20% of the hospitals, and 20% of the utility companies, and 20% of the water authorities aren't finished with their work? It may not be the end of the world, but I can't see any way of avoiding the conclusion that it will cause a series of unpleasant disruptions in our lives. The optimists will argue, of course, that 99% of the companies will get 99% of their

systems fixed; but as I've argued in many different articles and essays (including Y2K Projects: Deja Vu All Over Again), this would defy 30 years of software engineering history.

- Even if the large companies finish, the small companies won't. Several surveys in recent months have confirmed that nearly 75% of the small businesses around the world have not yet carried out any Y2K planning or remediation; even more amazing, 40-50% don't plan to do anything until something breaks on January 1, 2000. There are approximately 23.5 million small businesses in the United States; even if we assume that 50% start *and finish* their Y2K projects successfully, there's another 50% who won't start, and therefore won't finish. That's nearly 12 million businesses; if we optimistically assume that only 10% will be bankrupted by their lack of Y2K compliance, that's still sufficient to cause enormous disruptions in society. Among other things, the small businesses "feed" the big companies by supplying products and services; thus, if the small businesses fail, the big companies will experience a disruption.

- The same is true for small towns and municipalities across the country: approximately 50% have not yet done anything, and don't plan to do anything. Thus, even if one assumes that Y2K work is finished successfully in New York, Los Angeles and all the other major cities (which requires an enormous amount of optimism in itself!), there are thousands of small towns and counties across the country who may discover that their fire engines don't work, their 911 emergency communication systems don't work, and their water systems don't work.

- The same is true for small countries around the world, especially the developing nations who lack the money and the technical expertise to repair their own systems. One of the more recent examples is Russia, which has suddenly discovered (after denying the problem for months) that it needs $3 billion to fix its Y2K problems.

- Even if the big companies finish their Y2K remediation efforts, they're going to experience a rash of glitches and bugs. The emerging data in this area indicates that even fully-remediated and fully-tested software is likely to have an average of 450-900 defects (bugs) for every million lines of code. Some of these glitches will appear during 1999, as the "look-ahead" logic encounters Y2K dates; but most of them will hit on or about January 1, 2000.

- Optimists might assume that all of these bugs will be fixed within a matter of hours or days; but if the problems involve non-compliant suppliers, or non-compliant embedded systems, then it could take weeks or months to find a compliant replacement. Whether the organization can continue providing products and services while it's waiting for the replacement will depend on how good a job it has done with its contingency planning.

- Not all of the Jan 1, 2000 Y2K bugs will be discovered right away, and not all of them will occur right away. Some bugs will occur on Jan 1, 2000 but they won't have any visible impact; instead, they'll clobber a database record that might not be accessed for another month or two. Some bugs will occur on February 29th, when leap-year processing is invoked. Some will occur on March 31st, when financial systems begin summarizing results for the first quarter of the year; and some might not be visible until December 31, 2000 when companies try to close their books for the year.

Similarly, I can't prove that my economic forecasts are correct. Indeed, I'll be the first to point out that I'm not an economist; aside from a few semesters of college-level economics that I took in the 1960s, I have no training or experience with which to make expert predictions that anyone else

should necessarily depend on. But I have to depend on my own common sense, judgment, and forecasts as I plan my family's economic future; and my "gut sense" of the situation suggests to me that there are several reasons to anticipate a major economic decline:

- Even without Y2K, the global economy is on shaky grounds; the recent financial turmoil in Brazil, Russia, Korea, Indonesia, and other parts of the world suggest that it may only be a matter of time before our own stock market and economy are badly affected. Stock market prices -- especially for the high-tech companies like Yahoo and Amazon -- seem to have no relationship to revenues, profits, or other fundamental economic values; even without the international turmoil, I think it's only a matter of time before the bull market runs out of steam and retreats sharply. To whatever extent these scenarios are possible, I see Y2K as a "catalyst", or as the straw that breaks the camel's back. Thus far, Wall Street has blissfully ignored the potential threat of Y2K to the global economy; if and when the financial wizards *do* pay attention to the problem, I think they (and the small investor) are likely to react even more abruptly than if they had steadily factored Y2K into their plans throughout 1997 and 1998.

- The word "depression" is almost a four-letter word today; it's one of those words we're not supposed to use in polite society. If "recession" is equivalent to "darn," then "depression" is a hundred times worse than "damn." But if we look at the state of Japan's economy for the past eight years, it's hard to imagine a term more accurate than depression. It's not a "mild" economic downturn, and it's not a short-term phenomenon. It has lasted nearly as long as the Great Depression lasted, and in many ways (e.g., stock market averages), it's almost as severe. If it can happen in Japan, I believe it can happen in the U.S. and Europe as well.

- If the technological disruptions discussed above *do* occur, I believe it will wipe out a great deal of wealth in the stock market, in small businesses, in the real-estate market, and in the banks (because of the bad loans they'll have to write off). Much of the "wealth" in this country exists only on paper (or electronic "blips"), and we've grown accustomed to seeing it rise and fall somewhat sharply every time the Dow Jones average rises or falls a few hundred points. But if *real* wealth disappears through bankruptcies, foreclosures, and bad debt, I believe that it will lead to an overall decline in the economy.

- Those who have warned about Y2K alarmists causing bank runs will usually admit that the entire fractional reserve banking system relies on the confidence of depositors: if all of us lose confidence in the banks, and if we all demand to withdraw our money, then the banks collapse. But on a broader scale, the entire economy relies on confidence; that alone is a major reason why Japan is in a state of depression, and we're in an economic boom. We feel good about the economy, so we spend our money to buy products and services; this creates jobs, which creates income, which leads to even more good feelings. But Y2K could change that, especially if it challenges two basic assumptions that citizens in all advanced countries have depended on: first, *things work*; and second, *providers of products and services are presumed innocent until proven guilty*. Think about it: we *assume* that the lights will work when we throw the switch, and that drinkable water will come out of the spout when we turn the faucet. And think about this: when was the last time you asked an airline to "prove" its safety before you put your life at risk by climbing into a hollow aluminum tube and allowed yourself to be shot through the sky at 600 mph? If Y2K provides tangible evidence that (a) things don't always work, and (b) companies can't always be trusted to tell us the truth about the safety of their products and services, then our consumer confidence disappears. If we believe that our personal economic future is imperiled, then we'll save our money (and not necessarily in a bank!) rather than

spending it.

Conclusion

Might I be wrong about all of this? Absolutely! Should you feel obliged to agree with my assessment? Absolutely not!! You're more than welcome to imagine whatever Y2K future makes sense to you. My only recommendation is that you do it now, and that you do it explicitly. You don't have to post your opinions on the Web, but you should be able to jot down some coherent notes that say, "Here's what I believe about Y2K, and here's why I believe it." If you honestly believe that Y2K isn't a problem, and if you can articulate your reasons in a way that makes sense to (and to your family members, too, hopefully!), then you can carry on with your current life-style, and ignore all of the rhetoric that you'll be seeing in the media for the remainder of 1999. But if your own assessment convinces you that Y2K could cause moderate or serious problems, *now* is the time to acknowledge it, so that you can begin making appropriate plans before it's too late.

Home Ed's books Cool books *American Programmer*

Y2K Economics 101

Jennifer Yourdon's Response to Mitch Ratcliffe's Rebuttal of my Y2K Outlook

Introduction and Background

A few weeks ago, I wrote an essay entitled "My Y2K Outlook: A Year of Disruptions, A Decade of Depression." It apprently struck such a chord on the part of Ziff-Davis journalist Mitch Ratcliffe that he wrote a two-part rebuttal; part 1 is entitled "Economics 1010100010010011 ..." and the other is entitled "Daily Fix: How might Y2K bit the economy". I'm not sure how much this dialogue is actually adding to our mutual understanding of the Y2K situation, but since numerous people have emailed me to ask my opinion of Mr. Ratcliffe's commentary, I responded in a separate essay which I'll be uploading shortly. But since the entire argument revolves around the impact of Y2K on the economy, I asked my daughter, Jennifer Yourdon, to offer her thoughts and opinions on the matter.

Before you read Jennifer's response, you should probably read my original essay and both of Mr. Ratcliffe's articles.

Ed Yourdon

Y2K Economics 101

By Jennifer Yourdon, Co-Author, *Time Bomb 2000*

Dear Mr. Ratcliffe,

Though I spend much of my free time at lunch scanning through the Year 2000 websites I have bookmarked in order to keep current on news and opinions, I typically do not participate in any of the on-line Internet debates on Y2K -- primarily because I don't have the time to spend my workday on non-work related activities. My full-time, non-Y2K related career keep me occupied enough.

I feel compelled, however, to respond to your recent essay. I understand you've have written some previous Y2K commentaries, which I have not seen, nor have I had the opportunity review and critique the second part of your essay. The current topic you've written about is an area in which I feel quite comfortable: I have a Bachelor's Degree in economics from Trinity College, and have taken both PhD and Masters level classes in economics and mathematics at NYU. I have worked on Wall Street since 1993, doing economic research for foreign exchange proprietary desks, building econometric models to forecast and trade foreign exchange, equity indices and interest rates at a hedge fund, and finally analyzing hedge funds and constructing hedge fund portfolios using portfolio optimization techniques, correlation studies, sensitivity analyses and risk/return research.

I won't comment extensively on your nitpicking of my father's Y2K essay except to say that his expertise in various areas of technology (which have changed and developed over time with the industry), not only Y2K, is generally undisputed. As he has admitted, and you pointed out, he is not a

trained economist. I am. Are you?

For simplicity, I have put quotes from your essay in ***bold italic***.

I am sure my father used the term "thoughtful people" to differentiate between those who have taken the time to think about/ research Y2K, and those who casually dismiss it, and wait for others to tell them what to do. I am sure my father would also label individuals as "thoughtful people" if they took the time to research their own risks and decided to do nothing -- he just wouldn't agree with them. I honestly do not think "***Ed wants you to spend now, on supplies, and to change jobs, redistribute assets...***"; he has always made it very clear that all he can hope for is that people do enough research on their own. In fact, I am sure my father has enough to do and isn't particularly concerned with what your plans are.

For the remainder of my comments, I'll focus on the section of your essay entitled "The Economic Argument", and will allow my father to rebut the statements made about his beliefs and opinions.

The economic argument

I agree that my father's analogy of throwing the monkey wrenches into the engine of the global economy is not one you would find in a classic economics text book. It is however, illustrative and self-explanatory. Anyone can understand the scenario that he was envisaging: no one Y2K-induced problem would be enough to spark a slowdown in the global economy, but several at the same time might be quite harmful. Indeed, I challenge you to build an econometric model forecasting the likelihood and severity of a Y2K-induced recession, when there are no data with which to do this. That is a key point: the Year 2000 problem has never happened before, nor has there been a similar economic event that we can use as a model. Finally, the 0.3% dip forecasted by *some* economists (not *all* as you imply) only takes into account unproductive spending being undertaken to remediate the Year 2000 problem. It does not take into account possible disruptions in business, credit tightening, business failures, layoffs, increased trade imbalances, power outages, oil shortages, or a decline in the stock market.

Actually, my father did discuss further his vision of an economic decline as severe in length and magnitude as the Great Depression. He specifically discusses Japan, which has been in recession (depression) since 1990. In Japan, the government deficit has risen to almost 10% of GDP, the stock market has declined by almost 75% from its peak, and unemployment has surged. His argument was that whether you call it a depression, or an extended recession, this is not an event relegated to the 1920s: Japan, once the economic powerhouse of the world (much like the US is now) has fallen sharply, and this could happen again in a developed country. You dismiss his arguments, which are that big companies are behind, small companies are behind, governments are behind, foreign companies and governments are behind. The likelihood of them all finishing and testing on time, at this late stage, is low. In addition, it is "yadda yadda" not "***yada yada***" ...you are obviously not from New York.

Your argument that there have historically been different rates of failure in different industries, and that where the failures occur is important, makes sense. I am sure my father would "***recognize***" this point also. In fact, I think that any person of average intelligence understands that business failures at certain levels in certain industries have a larger economic impact than failures in others. I question using 1992 business census data for corporate bankruptcies, though I understand these are the most recent census data available. However, I examined more recent payroll employment data, and found

the manufacturing sector has been adding very few jobs, while the service sector has been adding four times as many jobs. In fact, in 1998 there were approximately 300,000 (seasonally adjusted) manufacturing jobs created, and approximately 1,200,000 (seasonally adjusted) service jobs (Source: Bureau of Labor Statistics). In fact, in 1991 (the year before your census figures), there were approximately 1,000,000 (seasonally adjusted) service jobs created, or a full 20% fewer than in 1998. Meanwhile, in 1991 there were approximately 285,000 (seasonally adjusted) manufacturing jobs created, or just 5% fewer than in 1998. These figures show that the growth in service-related jobs (and presumably the health of the service industry) has increased quite nicely, while there has been a rather paltry increase in the number of manufacturing jobs created every year. I would argue that this trend would suggest that the Internet and technology inspired economy of the past seven years has caused a shift in the general health of the service and manufacturing industries, and possibly in their bankruptcy rates as well.

As you note, some large manufacturers "***have announced their intention to end business relationships with suppliers that cannot ensure deliveries in 2000.***" Have you seen any such event take place? Have you seen any large manufacturer publicly announce that they are now ceasing business with supplier ABC because it cannot ensure deliveries in 2000? Either it hasn't started happening yet, and/or large manufacturers are only saying that they will end some relationships to soothe/calm analysts who follow their stock. In addition, some manufacturers (General Motors comes to mind) have so many suppliers that they a) cannot realistically expect responses from all of them; b) probably have a few suppliers who are the sole producers of a few of the goods they require ("***the only flange maker for diesel engines***"); or c) work on a just-in-time inventory basis, so that an interruption in even one factory or plant can disrupt the whole manufacturing process, as happened in the GM strikes in 1998 and 1996.

It is correct that most economists refer to GDP, rather than GNP, when discussing the state of the economy, though you define both of them incorrectly. Gross Domestic Product (GDP) refers to the value of the goods and services produced within the boundaries of an economy during a given period of time, usually one year. GNP is the total market value of final goods and services produced by citizens and businesses of an economy in a given period, usually one year. Thus, if Coca-Cola produces bottled soda in Indonesia and sells them in Hong Kong, those figures would not be included in GDP, as the US economy doesn't directly benefit from this production. In addition, one term isn't "older" than the other. As economies have become more globalized, it has become more accurate to use GDP to describe the state of the US economy.

In fact, GDP is not one of the ten indicators that make up the Leading Economic Indicators (LEI) published by the Conference Board. The ten factors are: average workweek, jobless claims, factory orders- consumer goods pace of deliveries, contract and orders- plant and equipment, building permits, unfilled factory orders- durable goods, raw material prices, M2, stock prices, consumer expectations. This, by the way, is very easy to look up, I easily found it on my Bloomberg at work. It makes perfect sense that GDP is not included in the LEI because the GDP is not *leading*, it is *old news*. GDP is reported quarterly, and is reported approximately a month after the end of the quarter, so a full third of the data is from four or five months prior.

My last point on your GDP discussion: if a 1.0% decline in 5 of the 10 LEI indicators is a "***strong recession signal***", then why hasn't it been accurate in forecasting a recession? There obviously isn't a high correlation between the two events!!

There probably is no one "***compelling driver of cyclical activity***" at any one time. Certainly, the

importance of any one factor changes over time as conditions change. To argue that stock prices are not important to the overall health of the economy is ludicrous. Stock prices become even more important to the health of the economy after a long bull market, where individuals wealth has been multiplied several times over. Perhaps you have never heard of the "wealth effect." This is the phenomenon of wealth being created or increased as the value of investments rise. If you make $100,000 in the stock market this year, you just might consider changing your spending plans. Similarly, if you lose your entire net worth in the stock market this year, you also might consider changing your spending plans. This is how the stock market affects the economy! So you'll know I am not alone in this opinion, I excerpted just a few key sentences out of Federal Reserve Chairman Alan Greenspan's recent Congressional speech:

> "The outlook for spending continues to be obscured to some degree by uncertainties about the course of equity prices; a failure of these prices to match the outsized gains posted in recent years would contribute to some moderation in spending growth, especially by households.
>
> Along with the numerous other uncertainties that attend the outlook, an additional uncertainty is present this year because of the approach of the year 2000 and the associated Y2K problem.
>
> At the same time, the wealth of households recorded another year of substantial increase, bolstered in large part by the continued rise in equity prices.
>
> The rise in net worth probably accounts for much of the decline in the personal saving rate over the past few years, to an annual average of « percent in 1998.
>
> With wealth rising faster than income over the year and with consumer confidence remaining at historically high levels, households were willing to boost their indebtedness to finance increased spending.
>
> Federal Reserve surveys indicate that banks responded to the turmoil in financial markets by tightening standards and terms on new loans and credit lines, especially loans to larger customers and those to finance commercial real estate ventures. The tightening reflected the less favorable or more uncertain economic outlook as well as a reduced tolerance for risk on the part of some banks."
>
> *Federal Reserve Chairman Alan Greenspan*
> *Humphrey Hawkins Testimony February 23, 1999*

Just to summarize a few salient points: 1) the outlook for the economy after an unprecedented run-up in equity prices is uncertain, as consumers have been adjusting their spending plans as the value of their portfolio rises; 2) the outlook for the economy is uncertain given the unknown impact of Y2K; 3) a substantial portion of individuals' increased wealth in 1998 was due to the increase in equity prices; 4) as wealth rises, consumers are willing to save less, and even borrow, to spend more; 5) when the financial crisis occurred in the late/summer and fall, not only did it impact consumers' wealth, but it impacted financial institutions' willingness to lend to consumers.

To make this as clear as possible: if there has been a big increase in wealth because of a bull market, if stock prices fall, individuals often want to spend less, banks are often less willing to lend to people (who want to buy cars and homes), sales tend to fall at corporations, and layoffs are possible. These are all key ingredients to a recession. Perhaps you would like to send your argument about this to Mr. Greenspan.

When you quote Ms. Foster's claim that "***Federal Reserve policy is more important than stock prices in shaping the business cycle***", you've identified something that *can* be true, but often times isn't. Referring to Japan again, the Bank of Japan (the Japanese central bank) has reduced interest rates to a historical low of 0.15%, and that country is still in recession. The BoJ has been easing credit since 1990, and yet they are still in recession. When there are structural economic problems, a loose monetary policy often isn't the right cure. Steep rises in interest rates do not always cause a recession. Indeed, the Fed did tighten interest rates significantly (by over 300 basis points) in 1994, and yet the United States did not have a recession.

An economic shock has no underlying assumptions, as you refer to in your essay. An economic shock is what it sounds like: a shock, a surprise, an unexpected event. If it were expected, or assumed, one could make preparations and there would be no impact. (If the US had known about the coming OPEC crisis in the 1970s, they would have stocked up on oil!) The word shock is often used with supply, supply shocks often occur when there are shortages of key goods, like oil. One can argue that although we all know the Year 2000 is coming, it still has the potential to be a shock because we don't know what the ramifications will be. If the United States is unable to produce or import oil for two months, that will very likely be termed as a "shock" to most people, and there will be the usual effects: increased prices, recession, etc. If US consumers take an extra $50 billion out of their banks, this will not be a shock, as the Federal Reserve has already planned for this. Yes, this is a critical point: there are no accurate assumptions, and as far as we can tell, policy makers are not taking any steps based on any assumptions. Finally, my father's belief that Y2K could be the straw that breaks the camel's back (the idea which you said "***underscores the naiveti [sic] of commentators on Y2K, like Yourdon***"), makes perfect sense. It is an unknown event, which most of the country will not have planned for, which will indeed by an unexpected shock to most of the country. I guess you should include Greenspan, too, in your group of naïve "**commentators on Y2K**" as he also thinks the economic impact of Y2K is uncertain.

When you state that "***the Y2K problem is not one***" of the reasons that the economy might slow, you only refer to the fact that non-productive resources will be redirected to productive resources once Y2K problems have been remediated. Here again you ignore the risks I mentioned earlier, including disruptions in business, credit tightening, business failures, layoffs, increased trade imbalances, power outages, oil shortages or a decline in the stock market. Other people have recently recognized these risks: The following was excerpted from yesterday's *Washington Post*:

> Senate Study: Y2K Risks Are Widespread
>
> By Stephen Barr
> *Washington Post* Staff Writer
> Wednesday, February 24, 1999; Page A1
>
> "The interdependent nature of technology systems makes the severity of possible disruptions difficult to predict. Adding to the confusion, there are still very few overall Year 2000 technology compliance assessments of infrastructure or industry sectors. Consequently, the fundamental questions of risk and personal preparedness cannot be answered at this time," the draft said. "

"***...low interest rates, rising liquidity (as measured by money growth) and stock prices, and further***

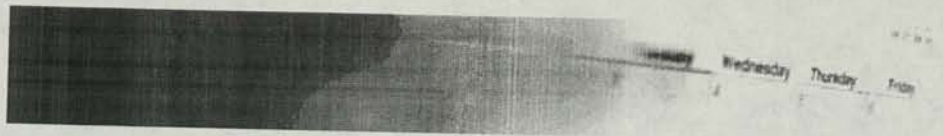
productivity will be the key drivers moving the US economy forward." Here you do reference a quote saying that the stock market is important for driving the US economy forward: so maybe you just don't think that the stock market can hurt the economy. If that is the case, and that is your belief, you are wrong. You also don't address any of the risks to the US economy. Here are some that Mr. Greenspan thinks are important: overvaluation of stock prices, the Russia crisis, the Brazilian devaluation, potential for a Chinese devaluation, continued weakness in Japan, continued weakness in other parts of Asia, a weakening Europe, increased socialist tendencies in Europe, increased protectionist beliefs in Europe, a ballooning US trade deficit, and oh yes, the Y2K problem.

What's most ironic about your penultimate paragraph, claiming that my father's analysis is "***narrowly***" focused, is that he and I have always kept our analysis very broad, as I have done in this rebuttal. We have always claimed that the Year 2000 problem can affect the economy in a variety of ways: on the demand and/or supply side, through layoffs, increased credit tightening at lending institutions, supply shocks, stock market declines, slowdowns in business because of disruptions of business in key clients, reduced demand from foreign countries, etc.

Finally, in reponse to your last paragraph: your article discourages anyone from doing their own Y2K research, does not draw on enough, or even accurate, economic data or viewpoints, has a very angry tone that seems unjustified.

Finally, I will repeat what my father and I have always said and believe in: We encourage everyone to do their own research, and to make their own Year 2000 plans based on their own findings. Frankly, we have enough to worry about with our own families in the coming year without spending a lot of time worrying about others as well. We feel we are doing as much as we can by informing people about the problem, informing them of resources where they can find and undertake more research, and by letting them know what we think. We have certain views on this problem, and have taken certain actions based on those beliefs, but we don't expect anyone to blindly follow our lead *nor do we want that responsibility*. We have made that viewpoint perfectly clear by refusing to disclose certain details of our plans: it is none of your business, and your plans are none of ours! We don't want to be responsible for others making big or small changes in their own lives…everyone's life should be ruled by their own thoughts, beliefs, ethics, and views.

Jennifer Yourdon
New York City
February 25, 1999



(Computerworld 12/14/98)

The moral dimension of Y2K

Ed Yourdon

It's now late December, and you're on your way out the door, ready to enjoy the holidays before 1999 begins.

But it may be your last holiday for quite a while. For most of us in the year 2000 world, 1999 will be the year of living dangerously.

If you're working in a bank, an insurance company or a telephone company, your year 2000 project is supposed to be wrapping up its remediation efforts now.

You're supposed to be spending all of next year on testing. After all, your CEO is telling Wall Street analysts that year 2000 remediation is "substantially complete," and the chief financial officer has studiously avoided documenting any bad news in the Securities and Exchange Commission's 10-Q statement.

But from my travels across the U.S., I know that there are a lot of organizations where that rosy, optimistic outlook isn't accurate. I've listened to a number of year 2000 managers tell me, within the past few months, "There is absolutely no way we can finish in time."

The dilemma of the doomed year 2000 project, or the

production before testing is finished, will force some of us to make some difficult ethical decisions next year. We can't abdicate responsibility for the decisions that need to be made in those year 2000 projects, even if we don't have the equivalent of a Nuremberg Trial to pass judgment on those decisions.

If year 2000 turns out to be a minor hiccup, then the Nuremberg reference will be heavy-handed. But if year 2000 turns out to have life-and-death consequences, then many of us will face the most difficult moral and ethical decisions we've ever made. We'll be hearing lots of advice from our managers, political leaders, friends and families about what we should and shouldn't do. Rather than try to tell you what you should do about your year 2000 obligations, I suggest that you read the codes of ethics compiled by the two most prestigious computing societies in the U.S.: the Association for Computing Machinery (ACM), and the Institute of Electrical and Electronics Engineers. They're posted on the Internet -- at www.acm.org/constitution/code.html and www.computer.org/tab/seprof/code.htm.

They're too detailed to discuss at length here, but I'd like to list 13 of the 24 "moral imperatives" from the ACM code:

- Contribute to society and human well-being.
- Avoid harm to others.
- Be honest and trustworthy.
- Strive to achieve the highest quality, effectiveness and dignity in both the process and products of professional work.
- Acquire and maintain professional competence.
- Know and respect existing laws pertaining to professional work.
- Accept and provide professional review.
- Give comprehensive and thorough evaluations of computer systems and their impacts, including analysis of possible risks.

its consequences.

- Access computing and communica-tion resources only when authorized to do so.
- Articulate social responsibilities of members of an organizational unit and encourage full acceptance of those responsibilities.
- Manage personnel and resources to design and build information systemsthat enhance the quality of working life.
- Articulate and support policies that protect the dignity of users and others affected by a computing system.

The irony is that if we computing professionals had insisted on following that code of ethics, we might have avoided the year 2000 problem altogether. But that's water under the bridge; the only relevant question is whether we intend to pay any attention to them in the remaining 383 days before the Big Day. All of us consider ourselves moral and ethical people -- it's simply a matter of which ethics we choose to embrace. I may not be able to memorize the entire ACM list, but I don't think I'll be able to live with myself in the post-year 2000 era if I can't follow at least the first three moral imperatives.

What about you? I urge you to think about this now, because if delays and problems occur with your year 2000 project, the pressure to compromise your principles may increase as we move into 1999.

Yourdon heads the year 2000 service at Cutter Consortium in Arlington, Mass. His most recent book is Time Bomb 2000. His Internet address is ed@yourdon.com.

The Y2K Crystal Ball: What's Going to Happen on 1 Jan 2000?

An interesting pattern is emerging from the steady stream of phone calls and e-mail messages that I receive each day. Many of the messages and calls can be paraphrased as one of the following:

- How bad will Y2K be? Exactly what is going to happen on 1 Jan 2000?
- Are you more optimistic or pessimistic than you were a year ago?
- I just heard that Y2K guru X has changed his tune -- he's more optimistic (or pessimistic) than he was just a week ago. What's going on? Do you think he has been brainwashed by aliens, or threatened by the FBI, or has he simply become senile in his old age?

To avoid having to write the same email response over and over again, here are my generic answers to these three questions.

How Bad Will Y2K Be?

I don't know. Neither do you. Neither does anyone else.

Sorry, but it's about time that we all admitted that, notwithstanding our predictions and estimates and guesses and wishes, we really don't know what will happen on 1 Jan 2000. Commentators, experts, gurus, business executives, and government leaders may *think* they know what's going to happen, and they may construct an eloquent and appealing argument to support their prediction. Chances are that if it appeals to your predisposed opinion about Y2K, you'll like it; otherwise, it will make you angry.

Many people who contact me are obviously frustrated by this state of affairs. "What good are all these experts," one college student complained to me, "if they can't tell us what's going to happen?" Another email correspondent complained to me, "I don't have time to read all of the hundreds of articles, books, Websites, and newsgroup forums about Y2K. Can't you just summarize it for me, and give me a simple answer: is it going to be a disaster, or can I stop worrying about this Y2K stuff?"

Think about it for a moment: if someone summarizes, abstracts, and filters the hundreds of disparate articles about Y2K for you, you're going to get that person's bias or prejudice mixed in with the summary. And if that person has an "agenda" or a particular "spin" that he wants to put on the Y2K phenomenon, you're going to get that, too. I apologize if I sound cynical, but do you really expect the Federal Reserve system and the banking community to give us a pessimistic outlook on their Y2K progress, even if the situation really *is* pessimistic? For that matter, it's simply not realistic to expect *any* senior business executive to stand up and say, "We've missed every milestone on our Y2K project, and we underestimated the cost by a factor of five -- just like the IRS. Our programmers are burned out and demoralized; they've formed a conga line, and they're dancing out the door with their last burst of energy ... and the CIO just quit. There is no way on earth we're going to finish even our

mission-critical systems in time. We're doomed; you might as well sell your stock now."

But beyond this obvious point, there's something more important: people seem to want a crystal-ball prediction expressed in terms of an "either-or" outcome. They want someone to say, "Either Y2K is going to be the end of the world (TEOTWAWKI), or it's going to be a non-event. I'm convinced it's going to be a non-event, and here are the 27 reasons why..." But this is an overly simplistic way of looking at the future, and it doesn't help us make effective plans for coping with what, at this point, remains unknown.

Here's a metaphor. Suppose you plan to drive your car from your home to the office, and you want me to predict the outcome. I could say, "You'll either arrive safely, or you'll be killed in a fatal accident. I think you'll arrive safely, and here are the 27 reasons why..." For a 200-word newspaper article, or a 2-minute TV report that's absolutely obsessed with reducing everything down to a black-and-white sound-bite, this might be an acceptable way to summarize the situation. But there are obviously a number of other outcomes, and it would be much more helpful if I could say to you, "The most likely outcome is that you'll arrive at your office without any problems. But there's a moderate chance -- perhaps one in ten -- that you'll be involved in a minor fender-bender accident along the way, in which nobody is injured, but a few hundred dollars of damage is inflicted upon your car. There's a smaller chance -- perhaps one in 100 -- that you'll be involved in an accident in which you or the other driver will sustain minor injuries. And there's an even smaller chance -- perhaps one in 1,000 -- that you'll get involved in a really serious accident, in which your car will be destroyed, and one or more drivers or passengers will be sent to the hospital with major injuries. And, unfortunately, there is a tiny chance -- perhaps one in 10,000 -- that you'll be involved in a fatal accident that will kill you."

Armed with that information, you can then ask yourself the obvious questions: *how much risk am I willing to tolerate? What should I do to reduce the risk?* A teenager might ignore all of this information, and drive a convertible with no seat-belts, no airbag, no insurance, and no auto registration. A more conservative person would consider the risk-reward tradeoff of a seat-belt and an air-bag so obvious that he would happily spend the extra time and money to reduce his risk. Some might eschew a convertible, and drive a Volvo instead. And a few might be so fearful of the risks, especially in urban centers like New York or Los Angeles, that they would decide not to drive at all.

The big problem with Y2K, of course, is that we don't know how to quantify the various risk scenarios; we don't have the benefit of 50 years of statistics about automobile accidents and fatalities. That's unfortunate and frustrating, but I don't think it's an excuse to ignore the problem. After all, we deal with uncertainty in our lives all the time, and we do the best we can to make an intelligent decision that's compatible with our level of risk-tolerance. When I get in my car, I don't have precise statistical data about accidents and fatalities for my neighborhood; I have a "gut feeling" that incorporates not only the general information about traffic accidents, but also up-to-the-minute impressions about the weather, the road conditions, and unusual circumstances (e.g., it's New Year's Eve, and there may be a lot of drunk drivers on the road). I go through a similar risk-evaluation process when I consider taking a stroll through a potentially dangerous neighborhood in a strange city, or when I respond to an invitation to go hang-gliding or sky-diving or bungee-jumping.

When it comes to Y2K, I think most of us will agree that the "non-event" scenario is relatively unlikely; there's a reasonably good chance that we'll experience one or more minor disruptions, at the very least. Beyond that, we each have to do our own risk assessment: how likely is it that the disruptions might last a couple days, or a month, or a year, or longer? And how much are we willing

to gamble that our assessment might turn out to be wrong? *All* of these scenarios need to be considered and evaluated, not just the two extremes of non-event and TEOTWAWKI.

One last point about the crystal-ball assessment, and the decisions we make: I think they deserve to be private. I'm really tired of newspaper reporters asking me, "So, Ed, tell me: just how much food have you stockpiled in your house? How much cash have you taken out of the bank? How many Kruggerands have you bought?" The proper answer, I believe, is: *none of your business!* After all, nobody asks me how much money I have in my savings account, or what percentage of my annual income is set aside for savings and retirement; nobody asks me how much auto insurance I have, or how much life insurance I'm providing for my family. At least in North American society, those questions are considered an invasion of privacy. But why do we bother with a savings account or insurance policies, if not to provide a "nest egg" for a rainy day? How is that any different from the decision that a few folks are making to stockpile some extra food in their pantry to cope with possible Y2K disruptions? While there might be emotional criticism about "hoarding" of food and supplies in December 1999 if everyone decides to rush to the grocery store at the same time, stockpiling ought to be a personal and private decision today — just like the decision to divert some of our disposable income to savings, rather than consumption.

Are You More Optimistic or Pessimistic Than You Were A Year Ago?

Here's what Jennifer and I wrote at the end of the preface to the forthcoming second edition of our *Time Bomb 2000* book:

> "To our many readers who have asked us, "What has changed since you first wrote the book in 1997?", we have three answers:
>
> - Another year has gone by, with little progress in large companies and the federal government, and almost no progress in small businesses and local municipalities. A year of procrastination and delay may be tolerable for long-term problems like global warming or the destruction of rain forests, but it's likely to be fatal for Y2000. Even 1997 was too late for most companies to start their Y2000 project with much hope for success; a company starting in 1998 has no choice but to use a triage strategy in order to keep at least its mission-critical systems running. Meanwhile, Wall Street remains, for the most part, sound asleep, despite the risks associated with Y2000 disruptions within corporate America, not to mention the risks associated with disruptions within the financial institutions themselves.
>
> - Research organizations, industry associations, consultants, and government agencies have helped to quantify the size, scope, and severity of the Y2000 problem. When we wrote the first edition of our book, we knew that there were a lot of banks, and a lot of utilities, that were at risk; now we know that there are approximately 11,000 banks and roughly 8,000 utilities in the U.S. This has reinforced our concern about the likelihood of a successful resolution of the Y2000 problem: given the dismal track record of software development

projects over the past 30 years, the statistical chances of 11,000 banks all finishing their Y2000 projects in time for the same inexorable deadline is vanishingly small.

- The concerns that we expressed in 1997, together with a few prominent economists and a larger number of computer geeks, have been validated and supported by a growing number of conservative, reputable politicians, industry officials, and corporate executives. When the President of the United States and the Prime Minister of England describe Y2000 as a serious problem, as they both did during 1998, it suggests that the subject should not be casually dismissed as an alarmist exaggeration. It requires only a modest effort to search the Internet to find somber warnings and/or dire predictions about Y2000 from the Chairman of the Federal Reserve System (Alan Greenspan), a former Secretary of Defense (Caspar Weinberger), and numerous Congressmen and Senators. Of course, none of the statements made by these officials constitutes proof that our assessment of Y2000 is completely correct -- but it does suggest that it's a legitimate topic of discussion and concern, rather than something reserved for the lunatic fringe.

Another common question that we've received from hundreds of readers during the past year is, "Are you more optimistic or pessimistic than when you first wrote the book?" From the comments above, the answer should be obvious: we're much more pessimistic. A year ago, we were dubious when high-level industry leaders and government officials publicly stated that Y2000 would be a "non-event"; as this second edition goes to press, we doubt that anyone in a position of authority would be willing to make such a statement...."

Procrastination seems to be most severe in the small businesses, and in the small towns. The optimists may sincerely believe that the *Fortune 500* companies will finish their Y2K projects on time, and that the Federal government will manage to at least finish the mission-critical systems in the most critical agencies. But recent surveys indicate that approximately 75% of small businesses *around the world* have not yet begun working on Y2K; here in the U.S., surveys indicate that approximately 40% of the small companies don't plan to spend any time or money on Y2K until the year 2000 itself, at which point they'll see what's broken. Similarly, recent surveys have indicated that approximately 55% of the mayors of local towns and communities believe that Y2K won't impact them; therefore, they're doing nothing about it.

One other dramatic change has taken place during the past year, and it bodes ill for Y2K: the financial crisis engulfing Russia, Korea, Indonesia, and several other parts of the world. Russia, known to some as Bangladesh With Missiles, has an official Y2K budget of *zero*. The country has no money for Y2K; it has no money for anything these days. Reports in late October 1998 indicate that food supplies have fallen to approximately a 2-3 week level, and that fuel stockpiles are also falling; chances are that Russian bureaucrats, business leaders, and citizens are far more concerned about the lack of food and fuel, as winter approaches, than they are about fixing a pesky computer bug whose consequences won't be felt for another 14 months (by which time they may be dead anyway). Similar problems confront the government and businesses throughout several Asian countries; it doesn't give

me much hope that Y2K projects will be given a high priority throughout the region.

What about the optimistic reports from U.S. agencies like the IRS and FAA? What about the reports from the banking community that the vast majority of its members are making good progress, and expect to be finished on time? In short, what about the *good* news? To paraphrase former President Ronald Reagan when asked if he trusted the Russians to live up to their promises after the nuclear disarmament treaty was signed, "Trust, but verify." With few, if any, exceptions, all of the optimistic news about Y2K has been *self-reported* status information; even if the people who tell us the good news are honest, sincere, and competent (an assumption we all have to evaluate for ourselves), they may still be wrong. Ask anyone who has worked on large, complex software projects: things often seem great until the system testing and integration begins. Even if dozens of serious bugs are discovered during testing, the project team and the project manager will exude confidence that the deadline will be met — right up until the day before the deadline, they'll earnestly tell you that the system-killer bug they just found is absolutely, positively, the *last* bug.

In this regard, the situation in late 1998 isn't any better or worse than it was in 1997. Serious testing -- end-to-end testing, and integration testing involving multiple firms, and multiple combinations of "supply chain" interfaces -- has not yet begun in any of the key industry sectors. All we know for sure is that 1999 will be The Year of Testing Dangerously.

Why Did Guru X Change His Outlook on Y2K?

Who knows? Who cares? Why does it matter?

Does anyone seriously think that God sent an angel down to have a little heart-to-heart chat with Peter de Jager, or Edward Yardeni, or Senator Bennett, or the multitude of analysts at the Gartner Group, informing them all that the Y2K situation really, truly, is going to be okay? Does anyone believe that the Devil sent one of his emissaries up from the fiery depths to have a little chat with Gary North, or Cory Hamasaki, whispering the bad news that Y2K really, truly is going to be TEOTWAWKI? Even if one of these individuals has indeed changed his outlook on Y2K, the fact remains: they don't know what will happen. Neither do I. Neither do you.

While there's nothing wrong with a consultant, analyst, guru, or Senator changing his opinion about the severity or consequences of Y2K, keep in mind that there might be several other explanations for the apparent change:

- The guru may have been misquoted by someone in the media. It happens all the time
- The guru may have been "trapped" by an interviewer who asks a ridiculous question, such as, "Do you think Y2K will be TEOTWAWKI, or a non-event? I have to squeeze all of this into 37 seconds on our evening news report, so I don't have time for any long speeches. Just pick 'A' or 'B'."
- The guru may have offered an opinion that sounds different from his previous statements, but couched in qualifiers and caveats -- e.g., "I used to think that Y2K was going to be a disaster. However, I'm more optimistic after speaking with the Grand Poobah of our country, and I think that we might have a chance of getting through this without major catastrophes if we all pull together, work 24 hours a

day, and make Y2K the top priority in the land."

- The guru's comments may have been taken out of context. This happens all the time on the Internet, especially in some of the Y2K discussion groups. Someone will post a message saying, "I just heard that Guru X has done an about-face on Y2K, and that he made a speech in Hog Jowls, Alabama stating that the Y2K problem has been greatly exaggerated..." Such a statement provokes dozens of responses, rebuttals, wild accusations, and general confusion -- until someone asks to see the transcript of the alleged speech.

It may be a sign of growing concern and panic that an alleged change of opinion is being interpreted by some folks as an X-files kind of conspiracy theory. "Maybe the goons from the FBI visited our favorite guru, and threatened to break his legs with a baseball bat," the Internet newsgroup participants mutter to one another. "Maybe it was the CIA. Maybe they threatened to audit his tax returns. Maybe they threatened to have Ken Starr investigate him. Maybe it will turn out that he had a menage-a-trois with Bill and Monica. Maybe..." Yeah, well, *maybe*. But I think not; it seems to me that if there were any conspiracy-theory forces at work, they would be aimed at the media, rather than a few noisy gurus.

One other thing to keep in mind: even gurus are human. They have good days and bad days. They occasionally make mistakes. They have fits of hysterical optimism, and bouts of lonely pessimism about Y2K. They visit one company that's doing a terrific job with its Y2K project, and they fall prey to the natural mistake of generalizing: "Maybe everyone is doing this well! Maybe it will all be okay!" And then they visit a government agency's Y2K project team, where the smell of death hangs in the air; with a simple "sniff test," and a few conversations with the peons who are doing the *real* work of the Y2K project, the guru confirms that the agency is doomed, regardless of what the top management says about the situation. And again, sometimes the guru makes the mistake of generalizing; it's not that difficult to conclude that *all* government agencies are incompetent, even though it's unfair and inaccurate .... well, somewhat inaccurate.

In any case, it's possible that a guru known for his pessimism may have been interviewed on a "good" day, when the sun is shining, when the news reports are optimistic, and when his best client has announced that they've finished all of their Y2K work, and have begun reassigning programmers to "normal" work. And it's possible that a guru known for his optimism may have been interviewed on a "bad" day, when he's suffering from the flu, is coping with a nasty hangover, is still grumpy about a fight with his spouse, and has been snubbed by a key government official who refuses to acknowledge the existence of Y2K.

Ultimately, none of this matters. The outcome of Y2K -- from a technological perspective -- is not going to depend on the opinions of gurus; it will depend on the hard work of several hundred thousand programmers, and a support staff of testers, documenters, bug-busters, end-users, and project managers. With only one or two exceptions, the gurus and commentators can't code; all they can do is stand on the sidelines and offer their opinion. That's fine, and the reason they *are* gurus is that people respect their opinions. But as suggested above, it's important for you to form your own opinion; then it won't matter to you whether your favorite guru has had a change of heart ... or a bad day.

Home | Ed's books | Cool books | *American Programmer*

Roleigh Martin, MA	marti124@tc.umn.edu (email)
Edina, MN 55436	http://ourworld.compuserve.com/homepages/roleigh_martin

Why are Year/2000 Projects So Difficult and Risky?

By Ed Yourdon

Year/2000 Journal

Volume 1, Number 6

November/December 1997

Reprinted with permission from Year/2000 Journal and Ed Yourdon:

subscriptions available at
http://www.y2kjournal.com

Because of its obvious importance and urgency, there's a tendency to approach a Year/2000 project with high levels of determination, energy and a "can-do" spirit. Naturally, we all

want our Year/2000 projects to succeed, the very existence of our enterprises may depend on that. But I'm concerned that our tendency to rely on the "power of positive thinking" may have obscured the underlying difficulty and risks with these projects. The fact that a CEO or CIO has publicly declared that "by golly, we will be finished with our Year/2000 projects by the end of 1998" is not enough to guarantee that it will be so. Management edicts will not repeal the law of gravity; and management edicts cannot erase the fact that a substantial percentage of software projects are canceled, delivered late, and delivered with numerous defects.

I have been queried recently regarding my "motives" for writing a Year/2000 book, and for making comments that might be regarded as unduly pessimistic by some: my motives are personal (protecting my family in the event of a serious Year/2000 outcome) and professional.

Professionally, I think I have something to contribute to the conduct of Year/2000 projects and planning activities, both at the local (corporate/project) level, and also at the national/strategic level. I've spent 33 years in the software field, starting as a junior programmer at DEC in 1964, and my credentials are an open book (24 of them, actually, for anyone to read). I don't claim to be the absolute authority on anything, nor do I claim that anything I say or predict can be proven correct with mathematical precision. But I have personal confidence that I know what I'm talking about when it comes to software engineering and software management, and I'm willing to put my reputation and career on the line in this area. The remainder of this article describes why I'm concerned.

Lousy Track Record

The issue of project management is one that has been mentioned in various writings about Year/2000, but it's often glossed over or dismissed-and yet it's the primary basis of my technical concerns about the risk of Year/2000 projects all over the world. Here it is, in a nutshell: the software industry has accumulated metrics from tens of thousands of projects (that's a literal number, not hyperbole) over the past 30-40 years, which tell us, statistically, that approximately 15 percent of all software projects are delayed (behind

schedule), and approximately 20 percent are canceled before completion. Statistics like this have been gathered and published over a long period of time, by well-respected software metrics professionals, long before Year/2000 was on anyone's radar screen.

One of the sources of information in this area is Patterns of Software Systems Failure and Success, by Capers Jones (International Thomson Press, 1996). But if you don't like the numbers from Capers, you can find almost identical numbers from the research carried out by Dr. Howard Rubin (head of the Computer Science Department at Hunter College in New York City, author of several world-wide software benchmarking metrics studies, and developer of the highly regarded ESTIMACS software-estimating product now marketed by Computer Associates) or Larry Putnam's company, Quantitative Software Management (Larry has also been in the software industry for 30 years, and is the coauthor of a book called Measures for Success: Delivering Software On Time, Within Budget (Prentice- Hall, 1992). The Stan-dish Group, the Gartner Group, the GAO, Scientific Amer-ican, and several other reputable sources have confirmed that our track record for delivering software on time is lousy, and has been for a long, long time.

Large Projects Suffer Worst

The situation is substantially worse for large projects. In the Capers Jones book referenced above, approximately 24 percent of all 10,000 function-point (FP) projects are finished behind schedule, and approximately 48 percent are canceled. For 100,000 FP projects, approximately 21 percent are behind schedule, and approximately 65 percent are canceled. If you're not familiar with function points: One FP is approximately equal to 100 COBOL statements; you can do the rest of the arithmetic. The point is that many of the large, mission-critical legacy systems that are the subject of Year/2000 remediation fall into this category.

Just How Late?

A related issue: just how late are the projects that are "delayed?" It turns out that the experience of the software industry, over the last 30-40 years, is that the average

software project is approximately six to seven months late. Again, the situation is much worse on the big projects: the 10,000 FP projects, according to the Capers Jones statistics, are an average of 13.8 months behind schedule, and the 100,000 FP projects are an average of 25.8 months behind schedule. Note that this doesn't include the projects that were canceled!

"We're Special"

For any I/T manager, or any CEO, who says, "Rubbish! Those numbers don't apply to us! We're special!" a reasonable response is, "Fine - then show me your statistics on the percentage of projects that are ahead of schedule, on time, delayed, and canceled. And show me your statistics on the average number of months behind schedule one should expect for a typical project in your organization." You'll find that in the substantial majority of I/T organizations, there are no such statistics. Or to put it another way, approximately 75 percent of U.S. organizations are at Level-1 on the Software Engineering Institute (SEI) process-maturity scale, which means they have no rational, well-defined, repeatable method of estimating their projects, nor do they keep statistics telling them whether previous projects succeeded, failed, or got delivered a year late. Whatever so-called estimates they produce are usually based on ad hoc guesses or negotiation, not estimating per se.

Edicts Don't Guarantee Success

My experience in the Year/2000 field for the past two years is that neither project managers, CEOs, or interested bystanders want to hear any of this. The reaction seems to be: "Oh, that's interesting" - and then they push it out of their mind and get back to their optimistic plans for Year/2000 projects. My response has been, "Wait a minute! Stop! Time out! On what rational basis can you conclude that your Year/2000 projects are going to behave any differently than the entire software industry has behaved for the past 30-40 years, especially since the Year/2000 projects are bigger than anything you've ever done, and you haven't even been able to formulate a plan or hire a staff to get started!" The answer seems to be, "Well, this time it will be different." But why? "This time, we really know it's important." But does that

mean your estimates will be any more rational or credible? "Well, this time we'll order the programmers to be more productive." Notwithstanding my opinions about management styles, I wish edicts would work in this case; unfortunately, 30 years of experience with software projects tells us that such edicts from high-level managers usually don't guarantee success. Indeed, the danger here is that the project manager and low-level techies will be bullied into delivering software that's missing essential functionality, or that's riddled with bugs. (You might want to take a look at my Death March book for more details about the consequences of such high-pressure projects).

Bugs Happen

Bugs, by the way, are another reason I'm pessimistic about the likelihood of a smooth Year/2000 transition in most organizations. Again, the pessimism is based on 33 years of experience: software project teams typically deliver "tested" software, which is put into production with an average of one bug for 100 lines of code (a certain well-known software company, for example, delivered a well-known PC operating system to the marketplace at the beginning of this decade with 5,000 known bugs in the code - known to the software organization at the time they shipped the product! This is not atypical behavior.). Some organizations are 10 times better than this dismal figure; a few are 100 times better; a very, very few isolated organizations like Motorola are practicing "six-sigma" quality techniques that can actually reduce the number of bugs to one in a million lines of code. But given the massive amount of software that's being Year/2000-remediated under pressure, without documentation, by programmers who are often unfamiliar with the original software - well, I can't see the rational basis for hoping or believing that we'll get through all of this work without a massive number of bugs that will take months, if not years, to exorcise from the systems. You can do the arithmetic any way you want, but given the mind-boggling amount of code that has to be modified across all the Year/2000 systems, a large rash of bugs is inevitable. Hopefully, most will be minor; some will be moderate and annoying; my concern here is that a few of them may turn out to be life-threatening.

Backwards Wishful Thinking

What's taking place on almost all Year/2000 projects is not estimating, but rather a form of "backwards wishful thinking." It starts as follows: everyone knows what the "ultimate" deadline is for Year/2000 - we can't negotiate or ignore that fact. Indeed, most organizations have arbitrarily decreed that their Year/2000 projects will, by golly, be finished on Dec 31, 1998. Not because anyone did any project-level estimating, or planning with Pert charts and Gantt charts, but simply because that's when management has decreed that things will be done. So if that's the deadline, then a rational project manager has to work backwards, leading to a train of thought that says, "If we have to be done by 12/31/99, that means we have to start testing by 12/31/98, which means that we have to be done with all of the planning and analysis by 6/30/97 - whoops! O my gosh, we're already five months behind schedule. Well, we'll make it up somehow by working very hard and convincing ourselves that we're not going to make any mistakes!" This is not a new phenomenon: we've been doing it for 30 years, every time management imposes an arbitrary deadline on project managers, because nobody has the guts to stand up and say, "Hell no!"

If you want to see a mind-boggling example of backwards wishful thinking, take a look at the published plans for Year/2000 projects by the major Federal government agencies. Note how many of them have an arbitrarily decreed deadline of 12/31/98 for finishing their work. Note also that the publicly stated schedule says, among other things, that 12 of the agencies are planning to devote either zero months, or one month, for the testing of all the Year/2000 remediation work they've done on their schedule.

And as I assume you've heard, four of the agencies were recently given "flunking" grades by Congressman Stephen Horn (a former university president, which may or may not be relevant) because they're so far behind schedule on the planning and assessment phase of their work, which is estimated by most Year/2000 experts to represent approximately five percent of the overall task.

The typical reaction to such criticism: "Oh, well, we'll catch up by working harder, and we won't make any mistakes, which is why we can get away with so little testing." Maybe

so, but such a statement contradicts the cumulative evidence of 30-40 years of work in the software field. Privately and off-the-record, the project managers (who get all the heat and the pressure from high-level managers when unpleasant news like this is made known) mutter to themselves, "We'll fudge the numbers on the next progress report, so that we can get these idiots off our back, and get some work done."

This is not a new phenomenon; this is what's been happening to large software projects since I got into the field, and probably long before that. And there is good reason to believe (based on first-hand reports from such project managers, which I'm not allowed to describe in detail because of non-disclosure agreements I've signed) that there are several of these situations underway in the government right now.

"This Time Will Be Different"

Using the available industry metrics, the statistically predictable fate of the 8,500 or so mission-critical systems (the number comes from the OMB folks) in the Federal government agencies is that we can expect approximately 1,250 of those systems to be finished late (by an average of one to two years, probably) and 2,100 to be canceled, except that cancellation is a rather grim option, and may not be allowed. These are not pleasant numbers, and one's natural reaction is to say, "That can't be true! This time will be different!" Maybe, but it's not as if we suddenly woke up with the advent of Year/2000 projects and said to ourselves, as an industry, "Well, we're tired of being lazy and unproductive and wildly inaccurate in our estimates; let's all unanimously agree to change our ways, starting today."

I've spent a substantial portion of my 33-year career writing books, providing training courses, and carrying out consulting work (as have several dozens of other, far more gifted and talented people than me) trying to accomplish this level of improvement that has suddenly been deemed imperative for Year/2000 projects. We've made incremental improvements in many cases, significant improvements in a few cases, and no improvement whatsoever in a few other cases. If the collective efforts of the methodology gurus and metrics gurus and project-management gurus in the

software field over the past 30 years has not been sufficient to eliminate the dismal statistics mentioned above, then I find it hard to believe that it's going to change suddenly in the next two years.

Paralyzed Senior Management

Maybe a miracle will occur and every one of those mission-critical projects will finish on time; maybe a similar miracle will occur in private industry. But the behavior, thus far, of both public-sector and private-sector organizations, belies this. Another interesting metrics-level observation in this area comes from Dr. Barry Boehm - whose Software Engineering Economics book (Prentice-Hall, 1981) and widely-used COCOMO software cost-estimating model, and numerous other accomplishments, put him near the very top of the software engineering guru list - and who has studied numerous projects that fail and succeed, in order to find out what makes them succeed or fail. His observation was that the largest single factor in the ability of a project to finish on time is when the project is actually started with real staff and a real budget, relative to when everyone knows that the project could be and should be started.

Common practice throughout the software industry is that an organization decides that it needs to develop project X by date Y; then they form a committee and argue about it for a few months; and then they assign a project manager, but don't give him a budget or staff; then they decide to change the objectives and requirements for the project; and then, after valuable time has been wasted, they finally get down to work, by which point the possibility of finishing the project on time has been reduced to almost zero.

This is exactly what's happening with many Year/2000 projects, because senior management is paralyzed by the cost, the risk, and the necessity of having to make some extremely difficult business decisions. I assume you've seen the statistics in this area; the Cutter Consortium recently published a survey in which only 53 percent of the respondents said they had developed a formal Year/2000 plan, 49 percent said they had explored the business consequences of Year/2000, and 35 percent said they had developed a Year/2000 triage plan.

The more you procrastinate on any large project, Year/2000 or otherwise, the greater the risk that you won't finish at all. (Alas, managers in many companies still haven't learned that you can't solve the problem by throwing hundreds of programmers on the project at the last moment, just as you can't produce a baby in one month by impregnating nine women. The failure of that approach was first documented by Dr. Fred Brooks, head of IBM's OS/360 project, in the classic book, The Mythical Man-Month (Addison-Wesley, 1975 - second, revised edition published in 1995).

Conclusion

All of the available evidence strongly suggests that we're not going to finish our Year/2000 projects. We'll certainly get 50 percent of them done, probably 80 percent, possibly 90 percent - but nowhere near 100 percent. Sure, some organizations will finish 100 percent of their systems (indeed, a few organizations have been Year/2000 compliant all along); but some won't finish any of their Year/2000 work, because they haven't heard about it yet, or because they haven't gotten around to funding and staffing it yet.

To ignore the significant likelihood that 10 percent or 20 percent or more of the mission-critical Year/2000 projects in this country won't be finished is irresponsible, in my opinion, not to mention imprudent. On a national level, sooner or later (probably later) someone at a high level is going to have to confront the significant likelihood that 900 or 1,000 or more of the government's mission-critical systems will not be remediated in time. Unpleasant news, to be sure, but to ignore it is something I'm not willing to do.

A final comment, regarding the TimeBomb 2000 book I am co-authoring with my daughter (available soon from Prentice-Hall). It was our assumption that various readers would find reason to disagree with with some of the worst-case scenarios we presented in one or two chapters, while others might disagree with the best-case scenarios (and I want to go on record as saying that I very much hope that the worst-case scenarios don't materialize, because it would cause enormous suffering and discomfort around the world). But we disagree with the notion that worst-case scenarios should be ignored altogether; as noted above, that seems

irresponsible, given what we know about the track record of the software industry.

We don't know what the outcome of the Year/2000 problem will be, nor does anyone else, not with mathematical certainty. There are many plausible scenarios, and what we tried to do in each of several important areas (banking, medicine, telecommunications, government, etc.) was to offer contingency-planning suggestions for four levels of Year/ 2000 impact: a disruption lasting two to three days; a disruption lasting approximately one month; a disruption lasting a year; and a disruption lasting a decade.

In most of our chapters, we concluded that it was very difficult to even imagine, let alone predict, a 10 year disruption - though Medicare, IRS, and Social Security are worth discussing. Even a one-year disruption seemed relatively unlikely in most cases; but the possibility of a one-month disruption is hardly radical. In any case, our premise was that responsible men and women should make their own decisions, based on their own assumptions, facts, guesses, informed opinion, or optimistic wishes. We continue to believe that's the most responsible thing we can do: let people decide for themselves.

It would be nice if the entire conversation could take place with scientifically verifiable "facts," but there are two problems with this: (a) we don't have a lot of time to gather these so-called facts about the actual progress of Year/2000 projects and then debate them, disagree with them, and finally publish them. If you're going to make fallback plans, the time to do it is now, with the best information you can get your hands on, not in December 1999. And (b), as you well know, most organizations have been advised by their legal department to avoid saying anything at all about their Year/2000 activities, for fear of litigation. We've been fortunate that the various government agencies have been "opening their kimono" to let us get a glimpse of what's going on inside.

I've had this kind of dialogue with many people during the past two years, and one of the most frustrating situations is when a CEO, or an I/T professional, or a layman, says to me, "Well, those are pretty impressive numbers you're throwing around - but it's not constructive. It doesn't tell us

what we should do to solve the problem." I sympathize with this attitude, for ours is a positive, optimistic, "can-do" kind of society: We like to define the problem, marshal our resources to deal with it - and then, by golly, follow Nike's advice and "just do it!" I certainly want the I/T community to work as hard as it can to fix as many systems as possible. The consequences will be far less serious if we finish 90 percent of the work than if we finish only 80 percent of the work. But it makes rational discussion more difficult when we decide, a priori, that it's non-constructive, negative, defeatist, and possibly even treasonous to say, "The problem cannot be solved. Not completely. Not anywhere close to completely."

I worry that if we continue along, during the next two years, with the optimistic "can-do" attitude we're going to get 98 percent or 99 percent or 100 percent of the Year/2000 work done, then we run the risk of not having done the necessary fallback and contingency planning that we ought to be putting into motion now.

That's an area outside the charter, authority, responsibility, and political power of the I/T department; it falls squarely on the shoulders of senior management in business and government. But thus far they're ignoring or avoiding it; as such, I consider President Clinton's vague, optimistic Year/2000 statement on August 15th substantially more harmful than any apocalyptic vision. (Editor's Note: See, An Open Letter to the President, page 78). As Peter de Jager observed in a conference last month, "Management will only support the notion of triage when they've absolutely given up hope of converting all of their systems." So far, it appears that they haven't done so.

In the final analysis, my plans and actions - which involves consciously and deliberately putting my reputation and career on the line - is not based on whether you agree with me, or whether any corporate CEO believes what I'm saying, or whether the government does anything constructive about Year/ 2000. More than any of these, my family matters most to me. I began writing the TimeBomb 2000 book with my daughter as part of a personal planning manual. That remains its primary purpose, and if it provides some assistance for others along the way, that will be an added benefit.

Having said all this, I wish you the best of luck with your own Year/2000 plans. Whether we're in agreement or disagreement about the details or the appropriate strategy to follow, the unavoidable reality is that we're all in this together.

About the Author

Ed Yourdon is chairman and co-founder of the Cutter Consortium. He has worked in the computer industry for more than 30 years and is also the editor of three software journals; American Programmer, Corporate Internet Strategies, and Application Development Strategies. Yourdon is author of more than 200 technical articles and has written 24 computer books. He is currently focusing on the Year/2000 crisis with particular emphasis on the interactions between business reenginering, object technology, and the design of industrial-strength Intranet/ Internet software applications. Cutter Consortium, 161 W. 86th Street - Suite 9A, New York, NY 10024. Voice: 212-769-9460, Email: ed@yourdon.com, Web: http://www.yourdon.com.

Last Updated 2/19/1998 11:02 PM

Return to home page

CORROBORATIONS

FORTUNE FEATURES

SECTIONS
Home
Business Report
· Street Life
· Subscribe Free
Smart Managing
· Ask Annie
Techno File
· Buyer's Guide
Fortune Investor
Personal Fortune
· Fast Lane
· Road Warrior
Small Business
Books
Special Issues
· Fortune 500
· Most Admired
Contents
· Features
· Columnists
· Archive
Forum

Search

RESOURCES
Free Trial Issue
Send Feedback
About Fortune

SERVICES
Marketplace
Ad Sections
Career Center
Conferences
Book Search

Pathfinder Privacy Policy

April 27, 1998

Industry Wakes Up to the Year 2000 Menace

Factories were slow to start fixing software that could cause shutdowns. They face a far more complex job than the rest of business.

Gene Bylinsky
Reporter Associate: Alicia Hills Moore

Crawling among pipes and valves at manufacturing plants around the world, technicians wearing radio headsets are relaying to companions carrying portable computers the locations of digital time bombs ready to go off on the Ultimate Midnight of Dec. 31, 1999. A massive hunt is on for the millions of computerized devices--machine tools, measuring instruments, computerized valves, and myriad other types of production equipment--whose software is tainted with the now infamous abbreviation "00." That bit of shorthand is how most programmers of yore, intent on saving memory space when it was thousands of times costlier than today, designated the year 2000.

Year 2000 bug hunters from Tava Technologies, a software fixer, check controls at the Climax Molybdenum mill in Colorado.
Photo: Mojgan B. Azimi

And thereby created the biggest screwup of the computer age. The bill for eradicating what has been called the Millennium Bug from the global economy's software has been estimated at $300 billion to $600 billion by the Gartner Group, a computer-industry research firm in Stamford, Conn. Even the lower figure would dwarf the federal government's huge bailout of the savings and loan industry in the early 1990s. Complete success is by no means assured, and the ultimate cost could be pushed past $1 trillion by hardware replacements and other expenses, including lawsuits that are already popping up. But failure to deal with the problem could cause telephones and lights to go dead and factories to halt, conceivably tipping the world into a serious recession.

Many computers go into the equivalent of catatonic shock when they read a year date of 00: The machines and devices they drive stop, print out wrong information, or malfunction in other ways. Averting a millennium mess entails more, however, than correcting year 00s to the full, four-digit 2000 wherever they are found. Innumerable interconnections between software programs must also be checked to ensure that a program inscribed with 00 does not, in the manner of a computer virus that plays havoc with your PC, corrupt a program that has been fixed.

The manufacturing sector has the most to worry about, for its year 2000 problems are more complex, widespread, and difficult to remedy than those in straight-forward computer applications such as accounting and finance. Worse, manufacturing corporations were slow to wake up to the enormity of the task they are belatedly tackling.

Unfounded gloom and doom? Not if you listen to Ralph J. Szygenda, chief information officer at General Motors, whose staff is now feverishly correcting what he calls "catastrophic problems" in every GM plant. In March the automaker disclosed that it expects to spend $400 million to $550 million to fix year 2000 problems in factories as well as engineering labs and offices. Or consider the words of Rob Baxter, Honeywell's vice president in charge of making his company's line of industrial control products "year 2000 compliant," to use computer industry jargon. From what he has seen among Honeywell customers, Baxter fears that "some plants will have trouble operating and will have to shut down. Some will run at a reduced scope. I expect considerable system outages during December 1999 through February 2000."

GM's problems are "catastrophic," says CIO Ralph Szygenda. Photo: Mojgan B. Azimi

Small wonder, then, that many plant managers and their bosses plan to stay close to their jobs over the three-day weekend when the millennium rolls in. Already they've had a foretaste of what could go wrong. A somewhat similar time problem--programmers' failure to account for the 1996 leap year--halted some production lines at the beginning of 1997, causing millions of dollars in damage. In simulations of the transition from 1999 to 2000, some factory robots, as well as computers that control electric power generation and transmission, stopped dead.

Other warning signs have appeared. By erroneously interpreting a 00 datum as the year 1900, a mindless computer at a food company directed workers to throw out perfectly good products. Many companies are looking to new enterprise resource planning (ERP) software to avoid such insanity. But a recent report warns that not all companies installing ERP can count on it for a safe journey into 2000.

The corrective task facing manufacturing is but one facet of a colossal undertaking by all of business. By diverting resources to "nonproductive endeavors," Federal Reserve Chairman Alan Greenspan has warned, the year 2000 effort could hurt the U.S. economy. Just for starters, according to the Gartner Group--where year 2000 problems make up the fastest-growing part of the organization's business--180 billion lines of software code will have to be screened worldwide, at $1 to $1.50 a line. The top 100 FORTUNE companies, Gartner figures, will spend $50 million, on average, to fix the year 2000 problem, vs. about $7.5 million for smaller companies.

Many of the programs in need of fixing are written in Cobol, once the dominant computer language for businesses. Cobol had no standard for writing year dates, which makes erroneous entries devilishly hard to find. Complicating the hunt, says George J. Kearsley, senior partner in MS Millennium, a Somerset, N.J., company working on year 2000 corrections, are the downright frivolous conventions some Cobol programmers created to label years. In their haste "to get the thing up and running," Kearsley says, programmers sometimes used whatever words came to mind, such as "time," "date," and in one instance "fliss." Says Kearsley, himself a former Cobol programmer: "It was the Wild West. 'Let's just do it. The boss is screaming to get it done.' "

Recalling "grayheads" Walter Dronov (seated, left) and James Neary (right) from retirement to supplement the skills of younger colleagues (back row), New Jersey's MS Millennium uses their knowledge in fixing old software.
Photo: Mojgan B. Azimi

Thus, software that searches automatically for erroneous year 2000 entries, with names like HourGlass 2000, Simulate 2000, and TicToc, can find only about 80% of them. The rest have to be located visually in a labor-intensive process, as do the interconnections and possible contagious effects of incorrect year 2000 dates on other programs. There's another problem. Since Cobol has not been taught for a decade, MS Millennium and other companies have been forced to recall Cobol "grayheads" from retirement to lend a hand.

The whole effort, in fact, is hampered by the general shortage of computer programmers. In some cases, big corporations have gone so far as to double the salaries of programmers working on the year 2000 and to offer bonuses as big as $100,000 if they promise to stay on beyond the turn of the century, until all transitional headaches are deemed solved. One company sends its year 2000 programmers and their families on three-day, all-expenses-paid vacations every eight weeks to exotic locales such as Marco Island, Fla.

As the fateful day draws closer, a disturbingly large amount of work remains to be done. A report released a few weeks ago by Forrester Research in Cambridge, Mass., says that on average, large corporations are only 34% of the way through the year 2000 job. The surveyed companies, on average, have completed 66% of the task of assessing the dimensions of the problem and the risks, but have made only 40% of the necessary fixes and have tested only 18%. The situation is worse in manufacturing, which is "behind business in general," says Bill Thompson, senior analyst at Automation Research Corp. in Dedham, Mass.

Manufacturing's task is compounded by the multiplicity of its computer programs. Below the layers of more or less standard software is a vast range of equipment run directly by built-in chips and programs, which outnumber those in the rest of business by a factor of ten.

Only about half of manufacturing's standard software was written in Cobol. The rest is a Tower of Babel, written in hundreds of tongues and added like onion layers to other software. Says William L. Swanton, director of research on plant operations at Advanced Manufacturing Research, a Boston consulting and data-gathering firm: "I'd be the most worried about the custom-built software created during the past 20 years."

There used to be very little off-the-shelf software for shop-floor use, Swanton explains, referring to types of programs such as manufacturing-execution, material-handling, and inventory-management systems. As a result, he says, "a lot of systems integrators and company programmers wrote applications themselves. Lord only knows what's buried inside those VAX minicomputers and other machines."

While software is available to hunt for 00 dates in programs written in Cobol and some other languages, few such tools exist for a vast number of so-called embedded systems in manufacturing. These are chips and programs, not readily accessible or even visible, that are integral parts of control and production equipment. Many must be decoded and fixed individually by hand.

Tava Technologies of Englewood, Colo., has one of the few software tools for automatically finding year 2000 errors in manufacturing's embedded systems. Among other things, Tava's program can read the "ladder logic" directing programmable logic controllers (PLCs). These simple, computerlike devices issue commands to factory equipment in the manner of a drill sergeant. Thousands of

PLCs dot factory floors, and all have to be checked.

Many embedded programs, however, can't be fixed at all because they are inscribed on silicon chips. In those cases, whole pieces of factory equipment, from time clocks to expensive computer numerically controlled (CNC) machine tools, have to be junked and replaced.

The severity of the year 2000 problem depends on a manufacturing company's age and size. Big, old metal cutters and benders like GM and Boeing face a particularly daunting task because of their hodgepodge of old production equipment and profusion of software programs. GM, Szygenda says, leads the world in the number of computerized systems. That doesn't mean that smaller metal cutters can relax. Technicians at Baldor Electric, the industrial-motor maker in Fort Smith, Ark., have been busily scrutinizing its equipment. On a recent day they checked out an Okuma CNC lathe despite assurances from its Japanese maker that it is year 2000 ready. Findings: The lathe is okay and luckily is connected to a computer system that uses no year dates.

At Baldor Electric in Arkansas, technicians found a big CNC lathe (foreground) free of year 2000 glitches.

Photo: Mojgan B. Azimi

Newer Silicon Valley highfliers such as Sun Microsystems, Varian Associates, and Tandem Computers, now part of Compaq, have less to worry about. The high-tech companies, for the most part, don't bend metal but assemble components supplied by outsiders. Companies like Sun and Tandem, furthermore, run assembly lines with their own computers.

Some newer companies, on the other hand, face a daunting task. At Catalytica's modern pharmaceuticals plant in Greenville, N.C., which employs 1,400 people, chief information officer Roger L. Dick lists 83 computer systems with about three million lines of software code. Within that code, says Dick, are 120,000 date references with potential year 2000 problems. The plant also boasts 138 automated production systems with 400 date references, plus 200 machines with embedded software, some of it in need of replacement to eliminate year 2000 glitches.

Modern or ancient, factories have some problems in common. Most manufacturing is driven by schedules and real-time demands for information processing just as severe as those in telecommunications and finance. The precision and interdependence of process controls in chemical plants, for instance, make a Rube Goldberg fantasy contraption look simple. Let a single temperature sensor in the complex chain of measuring instruments go cuckoo because of a year 2000 problem, and you'll get a product with different ingredients than you need--if it comes out at all.

Organization charts partly explain why manufacturing is late in addressing the year 2000 mess. Factory managers are generally isolated from the infotech bigwigs at corporate headquarters. For a while, many higher-ups thought the woes were largely confined to mainframes that run financial and telecommunications systems. They expected few, if any, problems down in the plant.

At first, factory types who warned of impending troubles were regarded somewhat like alarmist Chicken Littles, says Terry Landano, a manager grappling with the 2000 challenge at BASF Corp. in Mount Olive, N.J. Her company, a subsidiary of Germany's BASF, makes paints and plastics for the auto industry as well as fibers, vitamins, and chemicals. Landano recalls: "The top executives would ask: 'What, are you kidding? How could these two little digits be such a problem?' "

So for a long time manufacturing companies snoozed, including GM. When he arrived at the automotive giant a year and a half ago to take over the CIO job, recalls Ralph Szygenda, he was amazed "that most people assumed that the factory

floor didn't have year 2000 problems." Szygenda, with experience in manufacturing at Texas Instruments, didn't settle for assumptions. He shook GM out of its slumber by turning to outside companies such as Deloitte & Touche and Raytheon Engineers & Constructors, specialists in solving the problem, which sent in 91 experts to assess the automaker's situation. Supplemented by squads of GM technicians and programmers, these experts fanned out through GM's 117 facilities in 35 countries. What they found shocked even the factory-wise Szygenda.

"At each one of our factories there are catastrophic problems," says the blunt-talking executive. "Amazingly enough, machines on the factory floor are far more sensitive to incorrect dates than we ever anticipated. When we tested robotic devices for transition into the year 2000, for example, they just froze and stopped operating."

Szygenda quickly placed manufacturing facilities at the top of the list of the three "most dangerous" year 2000 areas at GM, followed by the company's supply base and the portion of businesswide software systems that supports production controls and logistic processes. Now, says Szygenda, "we're working feverishly and fast" to get the problem under control. All by itself, GM has two billion lines of software to check. The company is also retiring 1,700 obsolete computer systems.

Attacking the year 2000 problem has exposed another major area of vulnerability for GM: its 100,000 suppliers worldwide. Will all be compliant? Modern manufacturing's mastery of just-in-time parts delivery and business-to-business electronic commerce has created a beast that can bite it. Szygenda knows all too well how, on occasion, labor strife or a problem at a key supplier has shut down GM plants. "Just-in-time delivery has streamlined our supply chain to make it highly sensitive to any interruption," he says. "Production could literally stop at our plants if suppliers' computer systems are not year 2000 compliant.

He sketches the grim possibilities: "Let's say that a key sole-source supplier of brake valves shuts down as a result of a year 2000 problem. As a result, on day two, two plants that produce master brake cylinders and clutch master cylinders have to stop production because they don't have those valves. On day three, as motor vehicle assembly plants begin to run out of parts, production falls to about one-third of usual volume. By day four, all assembly plants shut down. And with no orders coming in because of the shutdown, hundreds of plants supplying parts to the assembly lines also shut down, from major engine plants to mom-and-pop subcontractors. That's the worst-case scenario--and yet it's a very real threat."

Surveying its suppliers last year, GM found plenty of cause for concern. The survey showed that awareness of the year 2000 threat was low among U.S. suppliers and even lower among those in Europe. One key global supplier didn't even know a problem existed.

GM is not alone in its worries. "The challenge of the year 2000 is that the dependency chain is so long, while you control only a small amount of it," says Jeff Pompeo, manager of information systems at GE Fanuc, a maker of industrial control devices in Charlottesville, Va. Libbey, a manufacturer of glass tableware in Toledo, takes the millennium transition so seriously that, according to CEO John F. Meier, his company is compiling a list of alternative suppliers--"mom-and-pop shops with pickup trucks."

With more clout than smaller companies, GM and the other two U.S. automakers, as well as truck producers, have banded together through the Automotive Industry Action Group to put pressure on suppliers. Unless they are in compliance for the year 2000, the organization is telling them, they won't be suppliers anymore.

Getting them up to snuff isn't always a simple matter. A supplier's computer system may be certified as year 2000-compliant by its maker, but hooking it up with systems from other vendors can bring trouble. "When you use one of our compliant products with, say, an Allen-Bradley CPU [central processing unit]," says GE Fanuc's Pompeo, "obviously we cannot warrant that it will be ready for the year 2000."

The majority of U.S. manufacturers have not even completed the year 2000 effort's first phase: plantwide assessment. Only after they take an inventory of software and embedded systems and estimate the impact of potential failures can they begin work on remediation, renovation, and finally testing and certification. Says Daniel Miklovic, senior manufacturing analyst at Gartner: "We advise clients to identify potential problems, correct those they can, require certification from vendors when they can't verify themselves--as will often be the case in an embedded system--and replace what they can't verify if they cannot convince themselves the problem is minimal. And document, document, document."

Hunting for glitches busies Foxboro's Richard Coombs (left) and Marie DeConto-Thomas.
Photo: Mojgan B. Azimi

Foxboro, a Foxboro, Mass., maker of industrial controls and automation equipment that's part of Britain's Siebe PLC, began ferreting out year 2000 time bombs a year and a half ago. It checked its manufacturing systems, including the hardware and software, as well as telephone systems, elevators, and security systems. "It seems that every time we had a meeting," says Foxboro's R. Michael Skelley, director of information services, "we were adding new devices to the list."

Foxboro has also been directing the year 2000 job at all the Siebe companies. The task, which among other things entails checking more than 8,000 PCs around the world, has occupied 1,000 programmers and technicians at the group's 230 worldwide sites. Meanwhile, Foxboro's customers have been checking whether the software it makes can handle the millennium transition. The S.D. Warren paper mill in Westbrook, Me., uses an older Foxboro control system to run papermaking machinery. It will need a year 2000 update, which Foxboro plans to have ready in plenty of time.

When it's available for manufacturing applications, software can speed the job of finding and fixing year 2000 glitches. "New year-2000 software companies are entering the market faster than almost any industry in history, at the rate of about 25 companies a month," says Capers Jones, chairman of SPR, a Boston-area software-research firm. Only a few offer software that can deal with factory problems. Among them are Raytheon Engineers & Constructors, Fluor-Daniel, and Peritus Software Services of Billerica, Mass., as well as the service operations of companies that sell industrial controls, such as Foxboro and Honeywell.

Still to be fixed: software in Ron Dyer's control room at Maine's S.D. Warren paper mill
Photo: Mojgan B. Azimi

One of the most imaginative and useful approaches comes from Tava Technologies. Its Plant Y2kOne software includes a database on 10,000 microprocessors, related control devices, and software from more than 1,000 vendors that is used on the factory floor. Among other things, Plant Y2kOne can check out software in robots, PCs, and PLCs; operating systems such as Unix, DOS, and Windows NT; and embedded software such as a program used to guide automated vehicles. After compiling an inventory of microprocessors and software, a client company can access Tava's database to determine whether the maker of each item can supply a year 2000 update--or whether the equipment should be replaced.

Tava will train a client's technicians to look for noncompliant items or send in its own team. Vice president Martin Fallon describes how a typical quest proceeds: "The team is on the plant floor. The team member with a headset says, 'I see an Allen-Bradley PLC.' His colleague, walking alongside with a laptop, scans down through the Tava list, finds Allen-Bradley and a list of PLCs, pulls it up, scrolls

down to the particular model number and checks on it, and it's added to the inventory of machines in this plant."

Back in the office, the technicians click on the Plant Y2kOne icon and submit the inventoried items to Tava via the Internet. The client can now see what's on a suspect list, vendors' statements, and Tava's own advice on the item. Tava's response on one device indicates that it can be kept: "Each intake node requires upgrade; system upgrade will take you approximately two hours per node." A different fate awaits a factory production monitor that Fallon brings up on his screen. The screen tells him it's a noncompliant product using the 00 date. Tava's advice: Contact the vendor and get a new model.

Software evaluation can be called up in the same way. In a demonstration, Fallon runs nearly 2,000 lines of code from a suspect program through a Tava "filter" and locates 22 examples of year dates. He looks at each line, and after finding the keywords related to date, lets the program determine if there's a year 2000 glitch and then changes the year digits from two to four. Tava also produces a printout for a client that shows noncompliant equipment in red, suspect in yellow, and "clean" equipment in green.

What the Plant Y2kOne program cannot do is calculate the relative risk inherent in all the interconnections among software in the machines. That still has to be done visually by skilled programmers. It typically takes three of them two weeks to sift through a million lines of code, which contain as many as 50,000 year dates. Tava's search tool has other limitations. Gartner Group's Dan Miklovic cautions that users should be "prepared for surprises," because such tools can't always read programs added on top of a PLC's ladder logic.

Once a company has identified all its year 2000 problems, it faces major decisions. Many devices can be fixed, but would it be better to scrap them? BASF's North American operations are replacing six different makes of personal computers in factories and offices with 12,000 Compaq machines at a cost of $60 million, including service contracts. Many of the old PCs, the company says, could have been fixed by replacing a BIOS (basic input output system) chip. But because that wasn't possible in all cases, it seemed better to standardize on the new PCs. "Old" is a relative term in the PC industry. According to analysts, some big-name PC makers were selling computers with noncompliant chips as recently as a year ago.

In some cases, fixing devices and software programs is downright unwise. It's a given in the software industry that new errors will be introduced in 7% of routine repairs. SPR's Capers Jones figures that the ratio of new bugs rises to 10% when the repairs are made on especially "tricky" kinds of defects that crop up in the year 2000 effort--for example, programs written in archaic languages.

While big investments in new hardware and software swell the cost of correcting the year 2000 problem, they can lift a company to new levels of efficiency. That's why the impact of the great undertaking is not all negative. "It's been called the cruelest technology joke of all time," says GM's Szygenda, "but at our company we think of it as a challenge that can be beneficial." Another plus: Many companies are finding out how inefficient their management of software has been and are planning major changes.

One of the most ambitious improvements is enterprise resource planning (ERP) software, which many companies view as a way to surge ahead of competitors in the 21st century. ERP drives all components of a company's business, such as defining materials that must be procured, scheduling shipments to customers, and handling financial and accounting systems. New ERP software costs more than fixing old programs. Picker International, a Cleveland maker of medical imaging equipment, could have patched its software for $20 million but is taking a $31 million plunge into ERP.

Putting in ERP systems also calls for extensive retraining of personnel. But chief information officers at Varian Associates, Tandem Computers, Catalytica, and Honeywell, to name just a few companies, hope not only to streamline operations

but also to cut out about 60% of the year 2000 remediation effort. All of which helps explain the sudden prominence of ERP software companies such as SAP, Baan, QAD, PeopleSoft, and J.D. Edwards. Advanced Manufacturing Research reports that ERP companies' revenues leaped 45% last year and have been growing at the same torrid pace in 1998.

But some ERP users may be in for unpleasant surprises. In a recent report, the Gartner Group criticizes the ERP providers for being slow in recognizing the year 2000 problem. Says Gartner: "The current state of the ERP market is not good." The report warns its clients that 34 of the 63 vendors it follows may not be able to bring all their clients' software up to readiness in time for the year 2000 rollover. Some ERP providers have criticized the report, insisting that they can do the job.

Many companies must wait until late this year to determine how much headway they are making through the year 2000 morass. In this booming economy, many manufacturers are running three shifts, seven days a week, and can't shut down for plantwide tests. The moment of truth will come for GM and others when they take advantage of holiday plant closings to do dry runs. Varian Associates' Tempe Electronic Center in Arizona will conduct a daylong test by running token production with a skeleton staff. Meanwhile, plant information technology manager William Benner and his staff of four are busily checking and rechecking all the production equipment.

The troubleshooters cannot relax because they know what's at stake. A host of developments have offered chilling previews of what laggard companies can expect when the year 2000 rolls in:

• Leap-year snafus damaged production lines when programmers failed to account for the extra day in February 1996. At a small U.S. manufacturer of industrial solutions that prefers to remain unnamed, production ground to a halt on Jan. 1, 1997. Before workers could remedy the situation, the liquids hardened in the pipelines, which had to be replaced at a cost of $1 million. That caused late deliveries and the loss of three customers. A similar leap-year oversight caused $1 million of damage at Comalco's aluminum refinery in Tasmania, when controls at all smelting-pot lines shut down, damaging five pot cells beyond repair.

• Computers misinterpreting 00 year dates are already issuing demented instructions. Companies don't issue press releases about such bungles and can't be named. But one consultant tells of a U.S. freeze-dried food manufacturer that noticed its warehouse inventory mysteriously decreasing. Reason: The computer system, mistakenly reading the 00 expiration date on the products as 1900, had ordered stocks destroyed.

• Dry runs are revealing a threat of cutoffs in vital services. When the Hawaiian Electric utility in Honolulu ran tests on its computer systems last year to determine if the year 2000 bug would have any effect, the systems quit. This meant that some customers could have lost their power or received it at a higher frequency, making clocks run faster and damaging or destroying valuable production equipment in factories. Communications systems could crash too. The U.S. Army's Materiel Command, testing its PBX telephone systems for the 2000 transition, found that they ran fine for three days after the turn of the year, until accumulated date errors shut down the whole network.

The first year 2000 lawsuits are already ringing up billable hours. In one widely publicized case, Produce Palace, an upscale grocery in Warren, Mich., last year sued TaecAmerica, a unit of Toshiba Group, which produced its computerized point-of-sale and inventory system. Produce Palace alleges that credit cards listing 2000 as their expiration date have caused the store's computer, installed two years ago, to crash more than 100 times, often for several hours, causing a loss of business while transactions were processed by hand. Toshiba says its policy is not to comment on outstanding legal matters.

GM's Szygenda thinks that "the year 2000 could easily spark the greatest litigation frenzy we have ever seen." Capers Jones of SPR projects $300 billion in damage

awards from lawsuits worldwide. Those early suits could serve a useful purpose by alerting corporate directors, CEOs, and top executives. They are sure to be named personally in year 2000 lawsuits, and to spend hours in the happy company of lawyers even if their companies have indemnified them against liability.

Companies are also on notice that they had better not mislead stockholders about their year 2000 risks. Starting this year, the Securities and Exchange Commission is calling on publicly held companies to disclose "anticipated costs, problems, and uncertainties." In its latest annual report, United Technologies said that gearing up for the year 2000 won't have a "material adverse impact" on its financial position. But it could be "adversely impacted," it added, "if suppliers, customers, and other businesses do not address this issue successfully."

"We're running with a sense of fear because we hear horror stories," says Keith Wettlaufer, CFO of A.G. Simpson Automotive, a big maker of auto parts in Scarborough, Ontario. He adds a little horror story of his own. The year 2000 testers at one of his suppliers, which he declines to name, couldn't get back into their plant because they forgot to check its security system for compliance. Says Wettlaufer: "The tentacles of this problem extend everywhere."

That's why New Year's Day 2000, a Saturday, will find legions of CIOs and other executives responsible for handling foul-ups passing up the chance to relax at distant beaches or ski slopes. Among those planning to be in the trenches are Leslie Schelp, who heads the 2000 project at Tandem Computers, and Varian Associates' CIO Donald R. McMorrow. Gartner Group's Miklovic volunteers where he won't be: "on top of a tall building or in an airplane."

GM's Szygenda recently told his company's suppliers: "No matter where you are in the journey, one thing will not change--the deadline. Dec. 31, 1999, will come regardless whether we're prepared or not. I'll bet there are quite a few CEOs out there that will be sweating as the big ball drops in Times Square." He's assuming, of course, that a year 2000 computer error won't cause the big ball to stall.

TECHWEB

Search CMPnet:

Search
Advanced Search

Free E-mail
Sign Up Now

Sections
News
Stocks/Finance
Web Development
Product Reviews
Company Profiles
Trade Shows
ActiveX/Java
Careers
Opinion
Humor

Support
Encyclopedia
Feedback
E-mail Newsletters

Featured Sites
Net Insider
NetGuide
Shareware
Games
Find An ISP
Horoscopes
CMPtv

CMPnet RESOURCES
Home
Site Map
Search
Buy Software
Buy Books
Ad Info
Sponsor Index
Link To Us

Free E-mail
Member Login

Password

April 06, 1998, TechWeb News

GM Awards Services Deal To CSC -- $85M contract for distributed computing and help desk reduces reliance on EDS

By Bruce Caldwell

General Motors Corp. last week carved out an $85 million chunk of EDS's business and awarded it to Computer Sciences Corp. The deal is the first designed to cut costs at GM by opening up pieces of the automaker's huge

EDS services contract to competitive bidding.

CSC won a 10-year, $85 million contract with GM's Locomotive Group, a $2 billion business unit. The contract covers help-desk and distributed computing services for 2,200 users, as well as engineering computing, applications development, and maintenance.

"This is a pretty stunning inroad into a previously closed, all-EDS domain," says Steve McClellan, a Merrill Lynch analyst. "And there are more pieces to come."

GM spun off EDS in 1996, about 12 years after acquiring the services firm, and began building an internal IT management team and strategy. EDS still gets the vast majority of GM's business, mainly via a 10-year services contract worth $3.6 billion a year. But GM plans to put as much as 25% of that business up for competitive bids, possibly as soon as 2001.

'Multivendor World'

The Locomotive Group deal means EDS and CSC staff will work together at the GM unit. EDS will provide WAN and mainframe computing services and carry out all year 2000 compliance efforts by year's end before handing over applications support to CSC. About 128 EDS technical staff will be offered jobs with CSC. "We will all have to learn to work together in this multivendor world," says Dana Deasy, CIO at the Locomotive Group.

Next, says Robert Chaffin, GM's finance director of information systems and services, GM may put out for tender parts of the $580 million business that EDS does annually with GM auto parts units Delphi and Delco. Chaffin says GM expects to put out to bid about $210 million this year, including the CSC deal, and the same in 1999.

The five vendors that competed for the CSC deal were each graded on pricing, service-level agreements, compliance with GM's request for proposals, metrics, quality of management, and ability to provide value-added solutions. CSC scored highest overall, says Deasy.

Deasy says cost savings from bids on the $85 million CSC deal have freed up funds for other technology projects. GM is awarding CSC a smaller deal for the first phase of a project to implement SAP Financials, for instance. GM also plans implementations of PeopleSoft's human resources applications and a product data-management application.

Gary North's Y2K Links and Forums

Summary and Comments

(feel free to mail this page)

Category: Introduction
Date: 1998-04-08 17:36:40
Subject: **State of Minnessota Exposes Six Y2K Myths**
Link: http://www.state.mn.us/ebranch/admin/ipo/2000/myths.html
Comment: This list of six myths appears on the state of Minnesota's Web site.

* * * * * * * *

IT'S ONLY A MAINFRAME PROBLEM. WRONG!

The Millenium Bug infects all kinds of computers (including personal computers), popular software applications, and it can strike computerized equipment such as traffic lights, medical lab equipment, elevators, and building heating and cooling systems. Even new technology is affected. Many recently purchased personal computers will not roll over to the new century without a Year 2000 fix! All technology must be evaluated.

IT'S ONLY A TECHNOLOGY PROBLEM. WRONG!

The Year 2000 Problem is a statewide business problem. If the technology problem is not addressed by government agencies and their business partners, suppliers, and service providers, it will impact government's ability to deliver the essential services that citizens expect and rely upon.

SOMEONE WILL INVENT A QUICK FIX. WRONG!

There is no overall quick fix solution to this problem because too many different kinds of computers and equipment are affected. Hard work and a systematic approach is required to find, fix and test all occurrences of the problem. Government agencies may be using tools to help speed up the process but there is no silver bullet fix.

WE HAVE UNTIL JANUARY 1, 2000 TO FIX IT. WRONG!

Computer information systems can fail now and some have failed already. The reason business transactions often project into the future. For example, government agencies grant licenses to citizens that expire at some future date. In addition, the Year 2000 Problem may cause severe resource shortages because it is being experienced on a world-wide basis and suppliers may not be able to meet the demand. The longer that government agencies wait to address the Year 2000 Problem, the more likely success will be negatively impacted by rising costs and shortages.

IT'S SOMEONE ELSE'S RESPONSIBILITY. WRONG!

At a minimum, we all have a responsibility to understand the issue, help find the problems, and assist Year 2000 project teams. Success depends on a collaboration of all agency personnel, including executive leadership, business managers, legal counsel, project managers, technical resources, facilities managers, contract managers, and human resources personnel.

MY AGENCY'S TECHNOLOGY IS COMPLIANT, SO THERE'S NOTHING LEFT TO DO. WRONG!

Testing and contingency planning are major components of all Year 2000 projects. Industry experts agree that testing is 50-60% of the effort and that all of 1999 should be devoted to validating that technology is indeed compliant. No

matter how well agencies have planned for Year 2000 conversions, failures may still occur as a result of unanticipated errors, or non-compliant service providers or business partners. Agencies must have well-documented backup plans for mission critical systems.

Link: http://www.state.mn.us/ebranch/admin/ipo/2000/myths.html

Return to Category: Introduction

Return to Main Categories Page

Return to Home Page

CBN News

- Top Stories
- Archives
- Anchors and Reporters

Editor's Desk

Christian World News

- This Week's Stories
- News Briefs
- Archives
- Broadcast Schedule
- About CWN
- Anchors
- Partnering Organizations

Contact Us

Support CBN

Countdown to Chaos: Preparing for 2000

March 23, 1998

Related Resources:
IRS 2000
Countdown to Chaos
Battling the Millenium Bug
Money Magazine
Gary North
Believers Church
Yahoo!

The race is on to meet the Year 2000 deadlines for fixing a potentially disastrous computer bug. And one big question is, will the federal government have its computer systems ready in time?

According to an official of the General Accounting Office, some government agencies probably won't be ready in time -- and others could miss the deadline if they have any last-minute delays.

Failure to meet the deadline could mean trouble for anyone who receives government checks, pays taxes, travels by airplane, or receives student loans.

National defense systems, including nuclear weapons, could also be affected if their computers aren't corrected in time. As of February 15, only 35 percent of the almost 8,000 computer systems labeled "mission-critical" were ready for 2000.

The government's most recent estimate for fixing the problem? $4.3 billion.

On Monday, March 23, Pat Robertson interviewed Edward Yourdon, a leading software consultant and co-author of *Time Bomb 2000: What the Year 2000 Computer Crisis Means to You*. Their interview, broadcast on *The 700 Club*, is reprinted here.

ROBERTSON: Mr. Yourdon, could I ask you before we get into this, could you just tell us a little bit about your background, because your book is explosive, to say the least.

YOURDON: Well, I've been in the computer field for about 35 years; I started in the mid-sixties. I've been a consultant for the last 20 years. I'm visiting companies all over the world to see how they run their computer projects. I've also written 25 books on software engineering, so I've certainly been in the business for a long time.

ROBERTSON: When I read your book, I was astounded, and I think when we look at what's involved, and maybe you could go through for the benefit of the audience just what the nature of the problem is and how extensive, because it seems to be a massive problem.

YOURDON: Well, it's fairly simple to explain. For the last 40 years, we've been deliberately programming computers to keep track of only the last two digits of the year, because everybody knows the first two digits are 19. This is 1998, the next year is 1999, and the year after that is 00. Unfortunately, the computers will generally think that it's 1900, rather than 2000, and as a result, will begin making a whole series of mistakes, ranging from fairly simple to possibly catastrophic. We've been seeing one example lately, with the credit cards that are coming out now with a "00" expiration date, which a few restaurants and stores think is a credit card that expired in 1900.

ROBERTSON: So, you're telling me that American Express, Visa, MasterCard.... I was reading a financial report from Paine Webber on Bank One, and they were saying that they thought that they hadn't come up to speed on this 2000 problem. How many of them out there have the problem at this moment?

YOURDON: Well, unfortunately, just about everybody. There are roughly 100,000 so-called mainframe computers -- those are the big computers that run the banks and

government agencies, and so forth. People have been working on that part for the last couple of years. Unfortunately, there are also some 300 million personal computers, not only in our homes, but also in our businesses. The IRS, for example, has 130,000 PCs. But the worst problem is some 25 billion so-called embedded systems -- those are the little microchips. And everything from your microwave oven to nuclear reactors and air traffic control systems and things of that sort. So it's an awesome amount of computer hardware and software that has to be fixed.

ROBERTSON: If you have one of those embedded deals -- say it's an elevator, or maybe it's a plant that says something about maintenance -- will that literally shut the plant down? Is that what could happen all over the world?

YOURDON: That's the danger. In the case of an elevator, for example, there are enough mechanical safety systems that you don't have to worry about the elevator crashing to the floor. On the other hand, if the elevator has some intelligence built into it with these microchips, it could decide to stop between floors, or simply come down to the ground floor and refuse to budge until the computer system's been fixed. Within the plants, the problem is that these embedded systems is that they are set up so that they check to see the last time they've been calibrated, or maintained, or lubricated, or what have you. And because they roll over from 1999 to 00. It might make them think that it's 1900; they'll get very upset, thinking it's been 98 years since they were last calibrated and shut the plant down. If that happens to be a utility plant, that means the lights go out.

ROBERTSON: You were very explicit about the IRS: we're going to do a piece later on about their preparation. I know the IRS spent about $4.5 billion to upgrade their system, and when it was finished, they said the computer software -- the logic they put into it -- doesn't work, and they're going to scrap the whole system. How prepared are they -- and this will affect the government in a major way -- are they prepared with their computer system to meet this problem?

YOURDON: The news report just a few minutes ago said there had been reports every three months or so on the status of the federal government, and so far only a third of the so-called "mission-critical" systems have been fixed. That same report card, or status report, predicted that by December 31, only two thirds of the computer systems will have been fixed. The IRS is certainly one of the more troublesome agencies, because of their previous track record, as you were saying. The FAA and several other ones are equally troublesome. And in addition to the roughly 8,000 mission-critical systems, there are another 66,000 ordinary computer systems whose failure will have some impact on all of us as well. I might also point out, by the way, that the government's budget for all of this has doubled within the last 12 months, which gives you some sense that they're still not sure just how big the problem is.

ROBERTSON: I've read all kinds of estimates; I've seen an estimate of $30 billion for the federal government, maybe $100 billion to $200 billion for the rest of society; what number is correct for the financial magnitude of this problem?

YOURDON: Well, as you say, there are numbers all over the place. The most common estimates at the moment are between $300 billion and $600 billion. Part of the problem is that we're still trying to discover how big and how pervasive the problem is. Actually, the larger expense, when you get into the trillions of dollars, is the litigation -- the lawsuits that are likely to occur both before and after the year 2000, from people who have been damaged or injured from all of this. In any case, it's going to be a massive number; there's no question about that at all, and of course it's all going to be non-productive in some sense. It represents money we could have been spending on more constructive things.

ROBERTSON: What does this do to the bottom line of corporations if they're suddenly diverting large amounts of money to fix their computers, and it's not going to give them any more productivity? That's bound to affect their earnings, which in turn could affect their stock prices.

YOURDON: You'd think so. Unfortunately, so far, Wall Street has been ignoring all of this. At the moment, most of the Fortune 1000 companies that are aware of this problem are diverting between 15 and 20 percent of their resources -- primarily people, but also their normal computer budget is also being diverted to this problem. So it's taking away

from other investments and new computer systems they could have been building. So as you say, it's beginning to hit their earnings. The SEC is now requiring companies to disclose how much they're spending on all of this. Last week, for example, Aetna Life Insurance indicated that they had spent $90 million on fixing this Year 2000 problem last year, and of course, they have more to go. Citibank is spending roughly $200 million last year, this year, and next year -- a total of $600 million -- which turns out to be only about 20 percent of their overall computer budget, but it's still a fairly significant amount of money, I think.

ROBERTSON: You know, I read a flyer by a guy named Gary North, which basically said, "I'm getting a home in the northern corner of Arkansas, because there'll be chaos in the cities and the trucks won't run, and the food won't be delivered, and grocery stores will break down, etc." And you had a few of those warnings here in your book -- what do you think? I mean, do you really think we're going to have some serious chaos in this country when this millenium turns over?

YOURDON: Unfortunately, I do. Let me just give you one or two numbers so you can see the magnitude of it. The three most fundamental things that we all depend upon to maintain some semblance of civilized life are utilities -- gas, electric, and so on -- telecommunications -- you know, this discussion we're having could not take place without the phone lines -- and banking. We're dealing with 9,000 electric utility plants in this country, of which *zero* are ready for the Year 2000 rollover at this point, 11 thousand banks, *zero* of which are ready at this point. Unfortunately, the track record of the software industry over the last 30 years is such that we can expect that 15 or 20 percent of all computer projects to be behind schedule, not just by a day or two, but by six months to a year. So I think we face the prospect of some degree of chaos -- blackouts, brownouts, possibly some degree of phone outages, possibly some bank failures. The GAO -- the Government Accounting Office -- said that there was a danger of as many as 700 banks failing just because of this Year 2000 problem. So I think there's going to be a period of chaos, if we're lucky, maybe only a couple of days, but I'm more inclined to think it may be more like a couple of weeks, maybe a couple of months, before things stabilize, after which, I think we're going to face a very, very severe recession. The economic impact of this is going to be very, very serious.

ROBERTSON: You also mentioned that certain countries might be isolated, that their phone systems might not work, or their air controllers might not work, so we might not have air travel between these nations: if an air traffic computer in the Slovak Republic doesn't work, will that turn off a Citibank computer in New York?

YOURDON: Well, that's the great danger -- not so much turning off the Citibank computer in New York, but sending garbled, non-compliant data. Obviously, the banking industry is global at this point, and that is one of our concerns. We're barely going to get 80 percent of our computers fixed at this point in this country, and that'll be problematic enough. Europe is being distracted by the Euro-currency project; that's keeping a lot of people busy. Africa and South America are sound asleep, and Asia is already very preoccupied with its current financial crisis, so it's more likely that Europe will be at the 60 percent level in terms of the percentage of computers they fix. Africa, South America, and Asia may be closer to the 50 percent level, so we may be forced to cut off -- to put a quarantine, on banking transactions to keep our own systems from being scrambled. As for air traffic and telecommunications, we may find that we're out of contact with the entire African and South American continents for several months after all of this begins.

ROBERTSON: All the major international currency transfers -- it's equivalent to about a trillion dollars a day -- is that system going to be compliant, do you think?

YOURDON: Well, if the Fed itself -- Alan Greenspan and others -- has said that 99 percent compliance is not going to be enough, it's very, very problematic and very risky. I should say that the banking community, at least in North America, is well aware of the urgency and the importance of all of this, and they're working, and they're working as hard as they can, but it literally is a race against time. We only have 645 days left at this point. And of course, the Federal Reserve system has great control over the banking system in this country and is putting enormous pressure on them, but they have no control at all over the rest of the world.

ROBERTSON: One last question: some people are hearing this -- what can they do? Should they move out of the stock market -- is there anything they can do to help themselves during this crisis? If we have a really severe depression, that means some kind of a stock market crash: what do you advise?

YOURDON: Well, there are four things that they've got to do: the very first thing you have to do is start doing your homework. This may come as a surprise to some people, but the news is out there. It's not on the front page, but you can find out about it on the internet, or by reading my book. The second thing you have to do is to take an inventory on how all of this might impact your life. It's quite different depending on whether you're in the northern part or the southern part of the United States. It's going to be wintertime in January of 2000. The third thing you have to do is to make your own assessment about how optimistic or pessimistic you are about all of this. Some people are optimistic and think they'll manage to fix all the computer systems; others, like me, who have worked in the computer business and have seen how all of these things can get messed up -- some of us are considerably more pessimistic. Then you have to decide what actions you want to take, including getting out of the stock market. If you're really pessimistic, you may get all your money out of the bank and bury it in the yard or under the mattress. In my case, I'm going to make sure I've got a month's worth of cash outside of the bank, and I'm hoping the banks will manage to get themselves back together again after a few weeks or possibly a month. But that ultimately is a personal judgement that everybody's going to have to make over the next 649 days.

ROBERTSON: I appreciate this wake-up; it's like Paul Revere riding along -- 2000's coming. Ed Yourdon's book *Time Bomb 2000* is available in bookstores around the nation, so thank you very much for being with us. We're going to keep you up to date on all of this, and on all of the compliance of various industries and agencies, so you'll know how things are going, because it's a major problem. It may not be as bad as some think, but on the other hand, it could be worse, so we'll just have to wait and see.

home about us on tv news ministry/teaching fact sheets audio

Gary North's Y2K Links and Forums

Summary and Comments

(feel free to mail this page)

Category: Introduction
Date: 1998-04-08 17:26:44
Subject: **Pat Robertson Meets Y2K**
Link: http://www.cbn.org/news/stories/980323.asp
Comment: Pat Robertson interviewed programmer/author Ed Yourdon on "The 700 Club." It was a long interview. It is posted on the CBN Web site (March 23). This exchange took place.

* * * * * * * *

ROBERTSON: You know, I read a flyer by a guy named Gary North, which basically said, "I'm getting a home in the northern corner of Arkansas, because there'll be chaos in the cities and the trucks won't run, and the food won't be delivered, and grocery stores will break down, etc." And you had a few of those warnings here in your book -- what do you think? I mean, do you really think we're going to have some serious chaos in this country when this millenium turns over?

YOURDON: Unfortunately, I do. Let me just give you one or two numbers so you can see the magnitude of it. The three most fundamental things that we all depend upon to maintain some semblance of civilized life are utilities -- gas, electric, and so on -- telecommunications -- you know, this discussion we're having could not take place without the phone lines -- and banking. We're dealing with 9,000 electric utility plants in this country, of which zero are ready for the Year 2000 rollover at this point, 11 thousand banks, zero of which are ready at this point. Unfortunately, the track record of the software industry over the last 30 years is such that we can expect that 15 or 20 percent of all computer projects to be behind schedule, not just by a day or two, but by six months to a year. So I think we face the prospect of some degree of chaos -- blackouts, brownouts, possibly some degree of phone outages, possibly some bank failures. The GAO -- the Government Accounting Office -- said that there was a danger of as many as 700 banks failing just because of this Year 2000 problem. So I think there's going to be a period of chaos, if we're lucky, maybe only a couple of days, but I'm more inclined to think it may be more like a couple of weeks, maybe a couple of months, before things stabilize, after which, I think we're going to face a very, very severe recession. The economic impact of this is going to be very, very serious.

Link: http://www.cbn.org/news/stories/980323.asp

Return to Category: Introduction

Return to Main Categories Page

Return to Home Page

Gary North's Y2K Links and Forums

Summary and Comments

(feel free to mail this page)

Category: Power_Grid
Date: 1998-04-08 18:07:04
Subject: **Texas Utilities: 100 Million Lines, Began in Mid-1996**
Link: http://www.tu.com/y2k/y2kissue.htm
Comment: Teaxs Utilities describes the Year 2000 Problem. Then it assures visitors that it began in time: mid-1996.

I am aware of no organization that has ever corrected, tested, and implemented a revision of 100 million lines of code. This includes firms that began earlier than mid-1996 (not many did). Let's hope TU makes it, along with all the other utilities.

Hope is not the same thing as "plan for personally."

Notice the phrase, "microprocessor-based systems." What does a large organization do with microprocessor-based systems? What are such systems? They are embedded chips. If TU has a problem with noncompliant chips and systems based on these chips, then there is far more to this repair than correcting bad code. Chips go out of production and cannot be replaced easily.

The final sentence is meant to be comforting: "The company is working hard to make a smooth, trouble-free transition to the next millennium." Too bad the labor theory of value isn't true.

* * * * * * * * *

TU has been addressing the problem since mid-1996. . . .

Could Y2K problems affect service?

TU is reviewing power plant controls and energy delivery system controls along with other company operations. While it is possible for microprocessor-based systems to have problems with the Y2K situation if not addressed before Year 2000, we are confident that this project will be completed well in advance of January 2000. . . .

Is TU monitoring compliance efforts of key suppliers and service providers?

As part of its compliance program, TU is contacting suppliers and service providers to obtain their support and determine their progress on this difficult problem.

This is a massive undertaking that will require a huge amount of work to accomplish. But TU began work on this project early and is well positioned to meet the challenge. The company is working hard to make a smooth, trouble-free transition to the next millennium.

Link: http://www.tu.com/y2k/y2kissue.htm

Return to Category: Power_Grid

Return to Main Categories Page

Return to Home Page

Gary North's Y2K Links and Forums

Summary and Comments

(feel free to mail this page)

Category: Power_Grid
Date: 1998-04-08 20:19:41
Subject: **No Standards for the Repair**
Comment: David Hall specializes in embedded chips. He has raised an important issue on Peter de Jager's forum: the absence of any agreed-upon standard for utilities and y2k. The threat here is that states will mandate conflicting standards. This could disrupt the power grid.

****

Reply-To:

From: "Dave Hall"

To:

Subject: Re: States and utilities

Date: Tue, 7 Apr 1998 19:51:39 -0500

Folks, I hope that there is some consideration for WHAT you can mandate at the state level. If Ohio mandates a certain Year 2000 standard for compliance and Iowa mandates another standard for compliance, we could end up with every utility being faced with meeting multiple standards for each state they operate in. And how about the companies that don't own anything but the power? How can you mandate that they, who do not own any equipment or facilities, meet some state-mandated standard? Since there is no universal standard accepted as being "the standard" used to make equipment and systems Year 2000 compliant, then there is no possible way, IMHO, for ANY agency, state, organization, smaller that the Federal Government to "mandate" standards of Year 2000 compliance for the electric utilities. For any other agency to try will lead to significant disruptions in any program already ongoing, and will cause many resources to be spent uselessly. Can you imagine what the consequences would be if Ohio mandated that every mainframe program used in Ohio had to meet the YYYY standard of Year 2000 compliance? Well, that's the consequences of trying to mandate utility compliance at a state level.

If any of you have any influence with state POCs, legislatures, local boards, PLEASE, PLEASE, try to explain this to them. Try and get them to NOT establish their own standards or definitions of Year 2000 compliance. Get them to petition, holler, scream, etc. at Washington and Congress. Only at that level can "mandated" standards of Year 2000 compliance be useful. For any industry, but especially the electric utility industry.

Return to Category: Power_Grid

Return to Main Categories Page

Return to Home Page

Gary North's Y2K Links and Forums

Summary and Comments

(feel free to mail this page)

Category: Government
Date: 1998-04-08 00:28:48
Subject: **Expert Warns Congress: Not Enough Resources to Meet Deadline**
Link: http://www.house.gov/science/harden_03-18.htm
Comment: On March 18, Dr. Michael Hardin testified to the House Subcommittee on Government, Management, and Technology regarding the rising cost of programmers. Governments must meet the wages offered by private industry.

Hardin referred to this as a "critical, and perhaps even fatal aspect of the government's ability to successfully deal with this massive problem."

This problem cannot be solved. The bids will rise; the supply will remain close to constant. We are facing an inelastic supply curve: more money will not bring forth significant new supplies of programmers.

We are about to see two dreams come true: Cory Hamasaki's (big bucks for mainframe programmers) followed by mine (a bankrupt, paralyzed Federal government).

There is now unconfirmed talk of the government drafting programmers. I love it! Can you imagine what a few thousand ticked-off mainframe programmers could do to every one of the Federal government's computer systems? How could the paper-pushing, responsibility-avoiding, clueless bureaucrats prevent the ultimate sabotaging of the New Deal, Fair Deal, Square Deal, Raw Deal known as the welfare state?

It's all over but the shouting. And the gnashing of teeth.

* * * * * * *

Mr. Chairman and Members of the Subcommittee:

I am pleased to appear before the Subcommittee to discuss the Year 2000 problem. I appreciate the opportunity afforded me to present to you, and the American people, what I believe to be a critical, and perhaps even fatal aspect of the government's ability to successfully deal with this massive problem. That issue is the possible inability of the Federal government to provide, acquire, or maintain sufficient programming resources to tackle the Year 2000 Problem in the short time remaining before January 1, 2000.

With only 653 days remaining, the prospects of achieving success appear ever more remote. As we move toward the Year 2000, our ability to achieve success becomes more dependent than ever on our ability to apply the necessary resources to the problem. Since there simply aren't enough programmers available to fix every system that may be affected by the Year 2000, the law of supply and demand takes over. . . .

Not applying sufficient resources today to fix the problem means that more resources will be required later to accomplish the fix in time. By increasing the demand for programmers, competitive forces will increase their salaries. So not only will we need more programmers, but the hourly cost for this talent will also rise. Every wasted hour today may cost us three or four times as much later for that same hour of work. . . .

Recently, Capers Jones, a noted authority on the Year 2000 Problem, estimated that the amount of labor needed to find, fix and test all of the Year 2000-impacted software was over 700,000 person-years. With less than two years left, there is no

way possible to complete the task for everyone. Many state governments have already felt the pinch. California, Texas, Missouri, Maryland, and many others have reported difficulties in retaining current programming staff or in hiring new staff. They cannot compete with the private sector salaries being offered. Private sector recruiters have resorted to offering "bounties" for trained programmers. And the problem is not isolated to the United States. In the United Kingdom, industry analysts estimate that all available programmers will be working on either the Euro conversion or the Year 2000 problem by April 1, 1998. Out of the 12,000 major UK companies needing assistance, there are only enough resources for 3,000. In Japan, there is an estimated shortfall of 200,000 programmers to fix the problem in time.

In conclusion Mr. Chairman, I cannot overstate the challenge that the Federal government faces in being able to assure itself of having the resources necessary to fix the Year 2000 Problem in time. Nor can I overstate the general industry consensus that the rising costs associated with hiring and retaining talented programming staff will impede the government's ability to achieve success in its Year 2000 battle. I urge the Subcommittee to focus itself on determining whether existing budget projections for the Year 2000 are still valid, and to quickly examine what measures can be taken to allow the government to compete on an equal footing with the private sector for the limited amount of resources available. The government of the United States cannot afford to find itself sitting on the sidelines watching commercial entities lure every possible resource out of the public sector, leaving the government hamstrung in its ability to deal effectively with the Year 2000 Problem.

Link: http://www.house.gov/science/harden_03-18.htm

Return to Category: Government

Return to Main Categories Page

Return to Home Page

Gary North's Y2K Links and Forums

Summary and Comments

(feel free to mail this page)

Category: Banking
Date: 1998-04-09 11:15:58
Subject: **International Banking Conference on Y2K: Crisis Mode**
Link: http://biz.yahoo.com/finance/980408/millennium_2.html
Comment: The Bank for International Settlements is the clearing house for the world's central banks. On April 8, it held a symposium on y2k. The press reports indicate that the BIS takes this problem very seriously.

The following is especially disturbing:

"Tom de Swaan, chairman of the Basle Committee on Banking Supervision and executive director in charge of banking supervision at the Dutch central bank, said it would be impossible for Group of 10 regulators (Belgium, Britain, Canada, France, Germany, Italy, Japan, the Netherlands, Sweden, Switzerland and the United States) to guarantee a bank or financial institute is ready for the millennium."

This is from a Reuters report (April 8).

* * * * * * *

BASLE, Switzerland, April 8 (Reuters) - Financial regulators and executives said Wednesday that urgent preparations were underway to avert possible network failures when computers tick over to the year 2000, a problem that could threaten global finance and lead to a worldwide recession. Speaking at a conference at the Bank for International Settlements (BIS) here, New York Federal Reserve president William McDonough noted that the world's payment systems, such as clearing organisations, settlement agents, securities depositories and others were ``intricately connected."

``An operational breakdown resulting from insufficient year 2000 preparations by any one of them may have an impact across payment systems," McDonough said.

A worst-case scenario could threaten the entire financial system, he added. . . .

In the United States, supervisors plan to have reviewed ``at least once" by June 30, 1998 the state of readiness for potential millennium problems at every institution they supervise. . . .

``If the current outstanding issues are any indication, we will have our hands very full," McDonough said.

Part of the danger lies in the fact that much of the world's money is booked or transferred electronically and the amounts involved tend to be staggering.

The New York Fed's daily payment settlement activities, for example, total roughly $2.5 trillion dollars, equal to about one-third the U.S. annual gross domestic product.

Contingency plans if computers fail because of a wrong reading of ``2000" are also being put into place. But upsets could involve everything from power grids to air travel and defence. . . .

Some 20 international banks on Wednesday also announced a new initiative to tackle the problems jointly, forming the Global 2000 Co-ordinating Group. . . .

Tom de Swaan, chairman of the Basle Committee on Banking Supervision and

executive director in charge of banking supervision at the Dutch central bank, said it would be impossible for Group of 10 regulators (Belgium, Britain, Canada, France, Germany, Italy, Japan, the Netherlands, Sweden, Switzerland and the United States) to guarantee a bank or financial institute is ready for the millennium.

Link: http://biz.yahoo.com/finance/980408/millennium_2.html

Return to Category: Banking

Return to Main Categories Page

Return to Home Page

Gary North's Y2K Links and Forums

Summary and Comments

(feel free to mail this page)

Category: Banking
Date: 1998-04-09 11:25:27
Subject: **Bank for International Settlements: Press Release on Y2K**
Link: http://www.bis.org/wnew.htm
Comment: The BIS is referred to as the central bankers' central bank. On April 9, it published a press release on its site. It can be found under "what's new."

What is significant in the report is the admission that y2k is a systemic problem. Also important is the admission of the need for contingency plans in case of a breakdown.

Problem: What is a realistic world contingency plan for the collapse of the banking system?

This document, for all of its subdued language, is basically a fire alarm. But no one knows how to put out the fire.

Consider these words:

"In particular, market participants from regions that have not yet vigorously tackled the problem should consider the need to invest significant resources in the short time that remains." . . .

"Failure of these organisations to prepare adequately and share information on their plans in order to promote effective testing could lead to serious disruptions in the world's financial markets." . . .

"Because widespread testing has not yet begun, the extent of the problems and the amount of remediation work that remains is unclear."

"Financial industry conventions and dispute resolution procedures should be developed to address the possibility of transaction failures. Additionally, contingency measures should be considered for the potential failure of key parts of the financial market infrastructure." . . .

"Even if an institution has verified and tested all its internally developed systems and applications, it will be affected by: the state of readiness of its vendors and third-party service providers; the public utilities upon which it relies, particularly the telecommunications and electricity suppliers; the infrastructures that it relies upon for its trading, payment and external information needs; and the counterparties and customers upon which its business viability rests. Moreover, the globalisation of financial and economic activity and the widespread use of information and telecommunications technology throughout the world has created various international interconnections and global interdependencies which add greatly to the complexity of the challenge."

* * * * * * *

Global Round Table on the Year 2000

On 8th April the Bank for International Settlements hosted a Round Table on the Year 2000, jointly sponsored by the Basle Committee on Banking Supervision (Basle Committee), the Committee on Payment and Settlement Systems (CPSS), the International Association of Insurance Supervisors (IAIS) and the International Organization of Securities Commissions (IOSCO). The meeting was attended by more than 200 senior executives (from 52 countries) representing a variety of public and private sector organisations in the financial, information

technology, telecommunications and business communities around the world. . . .

Sponsors' recommendations

1.Awareness of the seriousness and scope of the problem is high but varies considerably across markets and institutions worldwide. The sponsors believe that it is imperative that all market participants, and especially financial market supervisors, work to ensure that Year 2000 preparations receive the maximum senior management attention and priority, including at the board of directors level. In particular, market participants from regions that have not yet vigorously tackled the problem should consider the need to invest significant resources in the short time that remains.

2.Testing for Year 2000 readiness is the most critical and complex issue facing the financial industry. Because widespread testing has not yet begun, the extent of the problems and the amount of remediation work that remains is unclear. The sponsoring organisations urge market participants to explore ways to enhance the transparency of testing results.

3.The sponsors consider it critical that financial market supervisors around the world implement programmes that enable them to assess the Year 2000 readiness of the organisations and market infrastructures that they supervise. Further, it is important for supervisors to ensure that the risks related to the century date change are identified, properly communicated among market participants and appropriately managed in their jurisdiction.

4.The sponsoring organisations agree that the highest possible priority should be given to Year 2000 preparations by telecommunications and electricity providers in each national jurisdiction. Failure of these organisations to prepare adequately and share information on their plans in order to promote effective testing could lead to serious disruptions in the world's financial markets.

5.In order to achieve a greater degree of market transparency, the sponsors believe that the sharing of critical information on Year 2000 readiness by all market participants is essential. The sponsors encourage private sector efforts to develop standard questionnaires and frameworks for Year 2000 disclosure as these have the potential to provide clear means for measuring progress.

6.Financial industry conventions and dispute resolution procedures should be developed to address the possibility of transaction failures. Additionally, contingency measures should be considered for the potential failure of key parts of the financial market infrastructure. Further, it is recognised that national payment systems need to coordinate testing schedules in order to provide opportunities for end-to-end testing on a domestic and international basis. The Round Table sponsors welcome and support the initiatives taken by various industry groupings in this respect. . . .

DETAILS ON THE DISCUSSIONS AND CONCLUSIONS OF THE YEAR 2000 ROUND TABLE

Key issues raised at the Round Table

The presentations and discussions at the Round Table confirmed that the complexity of the issues associated with the Year 2000 cannot be overestimated. Moreover, every senior political and business executive should have realised by now that the issues pose a critical management challenge and have potentially severe consequences for the ability of business entities to continue operating through the transition to the new millennium. The many challenges include the need to address resource constraints, the need to share information on readiness (not withstanding legal issues), the coordination of testing schemes within and across markets, the heightened disclosure of vital information, the development of market conventions to deal with transaction failures and contingency planning for infrastructure problems.

Discussions at the Round Table confirmed that the Year 2000 challenge continues to require the unwavering attention of senior executives in institutions throughout the world. Organisations need to ensure that appropriate programmes have been established to address all the various readiness issues that can be expected to affect their business. The resource constraints (financial, human and technological) and the amount of time needed to remediate and test internal systems are compounded by the need to evaluate readiness and arrange for testing with counterparties and customers, payment, clearance and settlement systems, and various trading and information systems.

It is obvious that no individual IT user, individual institution, sector, market, or country is immune to the difficult issues presented by the Year 2000 problem. Even if an institution has verified and tested all its internally developed systems and applications, it will be affected by: the state of readiness of its vendors and third-party service providers; the public utilities upon which it relies, particularly the telecommunications and electricity suppliers; the infrastructures that it relies upon for its trading, payment and external information needs; and the counterparties and customers upon which its business viability rests. Moreover, the globalisation of financial and economic activity and the widespread use of information and telecommunications technology throughout the world has created various international interconnections and global interdependencies which add greatly to the complexity of the challenge. . . .

Testing for Year 2000 readiness may be the most critical and complex issue facing the financial industry in its effort to ensure that the various interrelated systems continue to function through the transition to 2000. Because widespread testing has not yet begun, the degree of uncertainty regarding the scale of the problems that will arise and the remediation work that remains to be accomplished is considerable. Further, the four sponsors urge market participants to explore avenues that will serve to enhance the transparency of testing results for the benefit of the remediation efforts of the industry at large. . . .

The significant dependence of most organisations on external product and service providers is another concern. The potential effects of a failing of the telecommunications or electric power infrastructure would have far-reaching consequences for global business. The sponsoring organisations agree that the highest possible priority should be given to Year 2000 preparations by telecommunications and electricity providers in each national jurisdiction because the failure of these organisations to prepare adequately and share information on their plans in order to promote effective testing could lead to serious disruptions in the world's financial markets. . . .

In the financial industry, in particular, it would be useful if conventions could be developed for the resolution of transaction failures, and contingency measures are considered for the potential failure of key parts of the financial market infrastructure. Further, it is recognised that national payment systems need to coordinate testing schedules in order to provide opportunities for end-to-end testing on a domestic and international basis. The Round Table sponsors welcome and support the initiatives taken by various industry groupings in this respect.

Link: http://www.bis.org/wnew.htm

Return to Category: Banking

Return to Main Categories Page

Return to Home Page

Gary North's Y2K Links and Forums

Summary and Comments

(feel free to mail this page)

Category: Banking
Date: 1998-04-09 11:50:21
Subject: **Yardeni's Speech to the BIS: Worldwide Crisis Looms**
Link: http://www.webcom.com/~yardeni/public/y_19980407.pdf
Comment: Ed Yardeni is senior economist for Deutsche Morgan Grenfell. He delivered a speech to the Bank for International Settlements at its April 8 conference on the Year 2000 Problem. He painted a frightening picture. His words will alienate those who dismiss y2k as a minor event:

"It is the ground-zero of the potential Y2K explosion. We all need to know if the products, services, information, orders jobs, incomes, and payments we depend on have been doomed by the triage decisions of those who provide them. If so, we might already be toast in 2000 and not know it in 1998 or even in 1999."

Most mainframe programmers regard such language as offensive, uninformed, overly provocative, and downright clear. They disdain any attempt to look beyond their cubicles at the society around them and draw conclusions regarding y2k's effect on the modern division of labor -- catastrophic. Those few who do see what's obviously coming make plans to move. This, too, bothers some of those who stay behind (for the moment).

Unfortunately for the programmers' mild-mannered, Clark Kent-like attempts to gain widespread y2k awareness in a calm, scholarly, guarded, and thoroughly professional manner, programmers do not get invited to speak to the BIS -- not that they've ever heard of the BIS, one of the most important institutions on earth. (I would say that it's the most influential secular international body there is, and has been for the last seven decades. What the BIS says, goes -- all over the world.) Nobody pays much attention to programmers -- not four decades ago when they established the short-sighted, digit-saving standard that now threatens our very survival, and not today, when they plead with management, "Gee, fellas, you guys really ought to give us more money to fix this." The same programmrers who did not have the Moxie to tell senior management until 1995 or 1996 that their computer systems are going to fail now get upset that people such as Yardeni use scary rhetoric. It's just not dignified. It's doom and gloom. It's extremism. It's, it's . . . motivational! Their motto is simple: "You may legitimately whisper 'fire' in a crowded theater, but only when half the people are unconscious from smoke inhalation."

I wish the document were in HTML. It's in a pdf file. You must have an Acrobat reader to access it.

* * * * * * *

Let's stop pretending that Y2K isn't a major threat to our way of life. There is too much at stake for such uninformed wishful thinking. Perhaps, the time has come to act as though we are preparing for a war. This may seem extreme and unnecessary. However, if we prepare for plausible worst-case Y2K scenarios, then perhaps we can avoid at least some of them. . . .

Securing Infrastructure.

Y2K "Sector Alliances" should be responsible for the Y2K campaigns in specific global sectors. The top priority must be to secure the supply of electricity worldwide. Other utilities, including water, gas, sanitation, and telecommunications, must also be secured. Contingency plans for rationing utility usage should be prepared. Other key sectors that may require a global "top-down" approach include government revenue collection and debt servicing, welfare

payments, farming, manufacturing, mining, transportation, distribution, retailing, banking, and finance. Y2K "Industry Alliances" should have the power to organize and execute a cooperative and collective battle plan among the world's key industries, including, for example, food, drugs, chemicals, energy, security brokerage and exchanges. . . .

Mandatory Y2K Holiday

The Y2K Alliance should consider requiring all nonessential employees to stay home during the first week of January 2000. Financial markets might have to be closed during this period. This global Y2K holiday would give IT personnel the opportunity to stress test their systems with a slow "reboot," rather than under peak load conditions. They could first test the integrity of basic utility services, especially electricity and telecommunications services. Then they could bring their own systems on-line in a phased sequence that can pinpoint weak links and either repair them quickly or take them immediately "off-line."

Emergency Budget

The Year 2000 Alliance Accord should require all participants to fund a Y2K Emergency Budget with an initial minimum balance of $100 billion. They should be prepared to provide much more, if necessary. The budget should be spent on both last-ditch efforts to repair or replace key computer systems around the world and to implement contingency plans once the weakest links have been identified. Conceivably, the funds may be needed to purchase strategic stockpiles of fuel, food, and medical supplies. . . .

The Governor of the Bank of England, Eddie George, calls January 1, 2000 "a day of judgment" and believes that the British government should freeze legislative and regulatory changes that would burden the computers of financial institutions already struggling with Y2K and the euro. In Canada, Prime Minister Jean Chretien is sending letters to all ministers and deputy ministers warning them that Year 2000 is the No. 1 priority and all other business is secondary. . . .

The division of labor could be radically upset by Y2K. This process is the very foundation of economic prosperity and progress based on the exchange of goods and services produced as a result of our comparative advantage. We all either thrive, or have the potential to do so, by producing the goods and services that we are especially endowed, qualified, or trained to produce. We exchange the fruits of our labor for the goods and services that are better made by others. IT systems have expanded the size of the markets and the opportunities for an even greater division of labor. Just-in-time manufacturing, outsourcing, and globalization are the most obvious modern extensions of the division of labor. Now imagine a world in which those IT systems either are impaired or completely fail. Suddenly, we may all be forced to do without goods and services that can no longer be produced for us by others. We can attempt to make them ourselves, but in most cases this will be impossible. If it is possible, the cost and time of doing so will be enormous.

There are no low-tech alternatives if our high-tech information systems fail in 2000. We simply cannot manually collect, sort, store, process, and analyze all the data we must have to support, let alone grow, our global economy. . . .

Only six countries have national Y2K awareness campaigns. A recent World Bank survey found only 37 out of 128 borrowing member countries said they were aware of Y2K.

Link: http://www.webcom.com/~yardeni/public/y_19980407.pdf

Return to Category: Banking

Return to Main Categories Page

Return to Home Page

Gary North's Y2K Links and Forums

Summary and Comments

(feel free to mail this page)

Category: Taxation
Date: 1998-04-09 17:15:41
Subject: **IRS Commissioner Warns of Total Economic Collapse**
Comment: Charles Rosetti, the head of the IRS, is predicting collapse if his staff can't get the IRS computers fixed. "The whole financial system of the United States will come to a halt." If he had a Year 2000 Web site, he would be accused of being a doom and gloomer and a scaremonger, too.

Rosetti's background is computers. His firm was involved in y2k consulting. He is the first Commissioner who is not a lawyer or a CPA.

Will the IRS get its computers fixed? Of course not. The IRS has tried to revamp its computers repeatedly over the last 30 years. The latest attempt cost $4 billion and took 11 years. The project was abandoned in January of 1997.

If the financial system comes to a halt in 2000, who will pay mainframe programmers to complete their unfinished work? It takes money to pay programmers. It takes banks. The banking system will not make it. Bank runs will close them. Programmers will not get paid.

This is the scenario that most programmers refuse to consider. This is why they reject all suggestions of a major breakdown in 2000. In the face of the evidence of universaal noncompliance and the tardiness of most major software revisions, they keep saying that most mainframe-based systems will make it, especially the financial system. They expect to be at the top of the food chain, that they will get paid when everyone else is frantically trying to earn an income. They're wrong. When the financial system falls, programmers will be fired, just like everyone else except local truck farmers. Yes, even the lawyers will have a hard time. (The collapse of the economy isn't all bad!)

It all hinges on the IRS and fractional reserve banking now. Get ready for a wild ride. Especially if you live in Washington, D.C.

Without the banks, the IRS can't get paid. Neither can state governments. Politics will revert to the county, which can collect taxes in cash. The county was the locus of primary political sovereignty when the U.S. Constitution was ratified in 1788. It will be again. Soon.

.

This is from USA TODAY (April 2).

****

WASHINGTON: Internal Revenue Service Commissioner Charles Rosotti said Wednesday he is so concerned about the year 2000 computer glitch that he wants congress to delay a massive restructuring of his agency. . . .

Preparing its computers for the next century will cost the IRS nearly $1 billion, Rossotti said. Although some new computers will be added, the lion's share of that money will go toward patching old systems.

"We've got to bring the IRS into at least the end of the 20th century," Rossotti said.

Perhaps the biggest obstacle, he said, is luring skilled computer professionals to the relatively low-paying public sector jobs. . . .

A: We've got to put things into categories so we can focus on things we need to

do right away, and the things we need to do longer term. The most compelling thing by far is fixing the computers so they don't stop working on Jan. 1, 2000. . . . If we don't fix [them], there will be 90 million people 21 months from now who won't get refunds. The whole financial system of the United States will come to a halt. It's very serious. It not only could happen, it will happen if we don't fix it right."

Return to Category: Taxation

Return to Main Categories Page

Return to Home Page

Gary North's Y2K Links and Forums

Summary and Comments

(feel free to mail this page)

Category: Taxation
Date: 1998-04-09 19:50:40
Subject: **IRS Loses Programmers, Offers 10% Raise**
Link: http://www.gcn.com/gcn/1998/March23/cov3.htm
Comment: Competition from the private sector has raised the vacancy rate in the IRS's IT staff from 4% to 8%. The IRS is offering a 10% raise.

This is from GOVERNMENT COMPUTER NEWS (March 23).

* * * * * * *

Over the next two years, IRS wants to spend $61 million on raises that agency officials hope will keep systems employees from defecting to industry. . . .

IRS will give about 1,000 programmers what the service called a 10 percent retention allowance for the rest of the year, he said. . . .

"We cannot not pay people and then expect high performance and increased productivity," Rossotti said. . . .

Programmers in Series 334 jobs at the GS-13 and GS-14 levels will receive the 10 percent salary hike. . . .

Rossotti told the subcommittee that the raises were necessary because private-sector systems workers receive annual salary increases of 12 percent to 15 percent.

To be eligible for a bonus, programmers must work at IRS headquarters in Washington.. . .

Generally, IRS has had a 3 percent to 4 percent vacancy rate for its information systems staff. But over the last couple of years, the rate rose to about 8 percent, IRS officials said. said.

Link: http://www.gcn.com/gcn/1998/March23/cov3.htm

Return to Category: Taxation

Return to Main Categories Page

Return to Home Page

Gary North's Y2K Links and Forums

Summary and Comments

(feel free to mail this page)

Category: Programmers'_Views

Date: 1997-07-17 00:00:00

Subject: **Yourdon: Should Programmers Quit and Leave Town in 1999?**

Link: http://www.cutter.com/ads/tyr0697.htm

Comment: Programmer-author Ed Yourdon asks the crucial question for all programmers involved in a y2k repair: If your employer isn't going to survive y2k, should you quit soon and take steps to protect your family?

This question is one of several Catch-22 questions associated with y2k. Any firm that plans on having its programmers dutifully working in 1999 to finish the y2k job will face defections: men who see that there is no hope for the company and no hope for the system known as the world economy. They will quit. They will head for safer places than the middle of a large city, where most mainframe computers are located.

* * * * * * *

But what if the problem cannot be fixed? What if January 1, 2000, arrives and half of your company's software has not been converted? What if your organization collapses as a result? At that point, where does your responsibility lie? The interesting thing about this is that almost every organization could have fixed its year-2000 problem if it had begun addressing the problem in 1995 or before. But if the year-2000 conversion team is just forming now, in mid-1997, then the conversion almost certainly won't be finished when New Year's Eve rolls around two years from now. And that part of the problem is the fault of senior management, which was too busy worrying about other issues to focus on the biggest software project of all time. . . .

Let me stop for a moment and address a basic point: Many software professionals believe that the year-2000 problem will be somewhat annoying, and somewhat expensive to fix, but they can't bring themselves to believe that it could be a major,

California if they really believe that they're going to wake up one day and find that the San Andreas Fault has finally ruptured, and that California is now an island floating in the general direction of Hawaii. "We've lived with plenty of earthquakes, and some of them have been pretty serious," these folks will tell you. "Someday, the Big One will hit, but I really can't believe it's going to happen this year, or next year, or the year after." . . .

What does all of this have to do with your job? Well, first you need to realize that a "denial of reality" may be taking place within your own organization today. Has your CEO or board of directors made a public commitment that all of the organization's systems will be year-2000 compliant, and that there is a detailed plan for coping with the organization's non-year-2000-compliant vendors, suppliers, customers, etc.? Do some arithmetic: If your company has 100 million lines of code in its application portfolio, then as of June 1, it would need to convert more than 100,000 lines of code per day, every day, in order to finish the job on time. Do you see the plans, the people, the tools, and the management commitment to make that happen?

If not, what will be the impact on your job? Chances are that your company will go into panic mode sometime in 1998, halt all of its development work, and assign everyone in the IT department to work on year-2000 conversions. When I say everyone, I mean everyone. Secretaries will be drafted into the testing effort, and the managers will be expected to begin writing COBOL code. Is that the kind of environment, with everyone putting in double overtime, that you want to work in? And if the year-2000 conversion isn't finished on time, who will be blamed? You can be sure there will be lawsuits; are you sure you'll escape the wrath of the lawyers?

If your company's year-2000 problems are very severe, what happens if it goes bankrupt? Will you be able to get another job? (An even more interesting question: At that point, will anyone want to admit that he or she is a programmer, or will it be a social stigma after January 1, 2000?) If you're out of work for six months, do you have enough money in the bank to support your family? . . .

Meanwhile, how many non-year-2000-compliant railroad and trucking companies will it take to disrupt the transportation infrastructure? While you're thinking about this, keep in mind another aspect of the lean manufacturing system -- the average grocery market, especially in urban areas, has to be restocked every 72 hours.

I don't have answers for all of these questions, and I spend a portion of each day wanting to believe that none of these crises will occur. But I can't find a way to deny the possibility that they could occur, not exactly in the way I've described, but as a series of domino-effect problems that ripple through society. And if my software experience allows me to anticipate some of this, then your experience should provide similar insights. Think about this, and try very hard to avoid the cognitive dissonance problem.

If the problem is anywhere near as bad as I think it could be, then you have to think very carefully about your loyalties and priorities. Will your employer get first call on your loyalty, or will it be your family and loved ones? On January 1, 2000, will you be at your keyboard, still converting two-digit year fields? Think about this now, while things are still calm. It won't be so easy two years from now.

Link: http://www.cutter.com/ads/tyr0697.htm

Return to Category: Programmers'_Views

Return to Main Categories

Return to Home Page

From: Bruce Webster <g8ubew@fanniemae.com>
To: WDCY2K@fanniemae.com <WDCY2K@fanniemae.com>
Date: Friday, March 27, 1998 9:03 AM
Subject: New GWU Lecture Series on Y2K

To the WDCY2K Group:

I was asked to pass this on, which I'm happy to do. As noted below, this is a new series of lectures on Y2K with a focus on group discussions and interactions. Both Stuart Umpleby (the organizer) and Jim Lord (the first speaker) are regular WDCY2K attendees. The usual disclaimers (i.e., this isn't associated with Fannie Mae or the WDCY2K Group) apply. ..bruce..

==========

The GW Center for Social and Organizational Learning has begun a lecture series on the Year 2000 Computer Problem. Our intent is to maximize conversations among those who attend. Below is an announcement of the next meeting.

TITLE: PERSONAL IMPLICATIONS OF THE YEAR 2000 COMPUTER PROBLEM

SPEAKER: Jim Lord

TIME: 6 p.m., Tuesday, March 31, 1998
PLACE: Room 404, GW Marvin Center
21st Street NW between H and I Streets

ABSTRACT:
The Year 2000 Computer Problem or "Y2K" is the worst technical blunder in history. At the very least, Y2K could cause serious disruptions across every aspect of our society. At the very worst, it might bring the greatest economic and social calamity the world has known since the Great Depression.

The purpose of this presentation will be to examine the most serious risks to society from Y2K and to identify strategies that might be employed by ordinary citizens to prepare for the effects of the Year 2000 Problem if these high-risk events actually occur.

Particular areas of exploration will include:
a. The banking system
b. Utilities
c. The "domino effct" or supply chain aspect of Y2K
d. Dependence on foreign computers
e. Y2K preparedness of the US government and
f. The potential positive aspects of Y2K

BIOGRAPHY:
Jim Lord is the author of "A Survival Guide for the Year 2000 Problem," a

practical, 270 page, consumer's guide to preparation for the Year 2000 Computer Crisis. It describes the effects of Y2K on all aspects of society and the economy. Particular emphasis is paid to government at all levels, employment, the financial sphere, and public services. Specific and detailed guidance is provided on how to protect yourself, your family, your assets, your job, your vital private data, and your personal safety. He is also the publisher of "Jim Lord's Year 2000 Survival Newsletter," which provides continuing updates on the progress of the Year 2000 Crisis. This bimonthly publication closely tracks the status of government and industry efforts to repair critical computer systems. It also reports on new strategies for protection against the effects of the Year 2000 Problem. See www.SurviveY2K.com. Jim Lord has been a guest on several radio interview programs and is a frequent speaker at conferences.

SCHEDULE:
5:00 p.m. Doors open and food available
6:00 p.m. Lecture and discussion
7:00 p.m. Small group discussions to share views and activities
8:00 p.m. Summaries of group discussions
8:30 p.m. Adjourn

DIRECTIONS: If you come by Metro, get off at the Foggy Bottom-GWU station. As you emerge at street level, walk two blocks straight ahead (east) to 21st Street and half a block to your right (south). On your right, facing Tower Records, you will see a ramp into the Marvin Center. Enter the building and take the elevator to the fourth floor. As you exit the elevator, go to your right and follow the signs to Room 404, to your left and straight ahead.

If you drive, there is parking beneath the Marvin Center. The entrance is on H Street between 21st and 22nd Streets. If this parking garage is full, there is a larger parking garage at the corner of H Street and 22nd Street.

RSVP:
Please send us your name, organization, address, phone number and email address so that we can make name tags, order sufficient food, etc.
The room we were able to get for this event is small, so sign up early.
We might have to limit the size of the audience.
Reply to Stuart Umpleby, 202-994-5219 or umpleby@gwu.edu

YOU MAY FORWARD THIS ANNOUNCEMENT TO OTHERS.
MY APOLOGIES FOR MULTIPLE POSTINGS.
==========

Gary North's Y2K Links and Forums

Summary and Comments

(feel free to mail this page)

Category: Domino_Effect

Date: 1998-04-13 10:20:30

Subject: **Global Positioning System: Another Bug**

Link: http://www.latrobe.edu.au/www/crcss/millennium.html

Comment: In early 1997, I posted a link to the U.S. Navy's site that announced the 1024-week rollback of the Global Posotioning Satellite System. Now there is an additional bug, one which will mess up commercial products that rely on the GPS to identify locations.

Most of the companies knew of the old bug; it was built into the design from the beginning decades ago. But this one may catch them flat-footed.

* * * * * * * *

GPS, the Global Positioning System established and run by the US Department of Defense, only became fully operational and world-wide in 1995, but is of ever-growing importance. The use of GPS is becoming vital not only for navigation, but for many applications such as precision farming, in mining, urban planning, and tracking of vessels, vehicles, and containers. News Flash Administrators of Joint Program Office that administers the GPS Program for the US Air Force recently announced the discovery of a bug in the last 36 hours of the current (GPS) millennium which will cause many receivers to reset to their start-up date. This start-up date may be the first date in the GPS millenium in 1980, or a later date in 1984, or it might be the date that the receiver was factory-tested or was first field-operational. This new GPS Millennium Bug, is in addition to the Y2K and the anticipated GPS Bug in receivers due to failure to properly anticipate the EOW roll-over.

Link: http://www.latrobe.edu.au/www/crcss/millennium.html

the year 2001.
The Australian Government support for Australian Space Technology, with its support for FedSat 1 through the CRC for Satelliet Systems, and its indirect support for ARIES will help Australia to new opportunities in space in the next millennium.
But the question remains.
Which of the two Australian scientific satellites about to be launched will truly be the Millennium Satellite?

	2000	2001
Australia in Space a History	ARIES Australian Resource Information and Environment Satellite A low earth orbiting (LEO) microsatellite ARIES will be equipped with sensors to scan the earth over a range of frequencies, to evaluate ground cover and type, and, especially in arid regions, map rock types and geological data.	FedSat 1 Australia's Scientific satellite launched to celebrate the Centenary of the Federation of Australian states January 1, 1901. A low earth orbiting (LEO) microsatellite FedSat 1 is to be the first satellite to be launched by the Australian Cooperative Research Centre on Satellite Systems. The CRCSS aims to foster Australian satellite design, communication, control, and operational engineering skills. FedSat 1 is to use the recently established network of GPS and Glonass satellites to study the ionosphere and atmosphere. (See ATMOZ-GPS)
	The well known Y2K or millenium bug arrives too early to affect FedSat 1 or ARIES. At 0000 hours, UTC, 22nd August 1999 there is the GPS EOW roll-over.	
CRCSS Home Page	NEWS ABOUT SATELLITES	Quiz on Satellites and GPS

Matilda

Enquiries and Comments to the Webmaster Dr Harvey A. Cohen

New GPS Millennium Bug Discovered

GPS, the Global Positioning System established and run by the US Department of Defense, only became fully operational and world-wide in 1995, but is of ever-growing importance. The use of GPS is becoming vital not only for navigation, but for many applications such as precision farming, in mining, urban planning, and tracking of vessels, vehicles, and containers. News Flash Administrators of Joint Program Office that administers the GPS Program for the US Air Force recently announced the discovery of a bug in the last 36 hours of the current (GPS) millennium *which will cause many receivers to reset to their start-up date. This start-up date may be the first date in the GPS millenium in 1980, or a later date in 1984, or it might be the date that the receiver was factory-tested or was first field-operational.* This new GPS Millennium Bug, is in addition to the Y2K and the anticipated GPS Bug in receivers due to failure to properly anticipate the EOW roll-over.

Y2K Computer Millennium Bug emerges January 1, 2000

Back in the sixties and seventies, and even beyond, many computer systems used just 2 digits to store year data, so only 99 years can be consistently stored. at the end of the first century lying within the Information age, at January 1, 2000, the problem first arises. Its called the Millennium bug, but really its a century bug.

EOW Satellite Millennium Bug emerges 132 days earlier

Satellites are potentially afflicted with a genuine millenium bug, which can emerge on August 22, 1999. This Millenium Bug is based on a digital thousand, 1024, and its also called the GPS EOW Rollover Bug.
GPS system time counts weeks from midnight 5-6 Jan 1980 in modulo 1024 (0-1023) -- a digital millenium of weeks.
The data packets from GPS SV (space vehicles) include all time information, including the week.
At 0000 hours, UTC, 22nd August 1999 the first GPS EOW roll-over occurs.

For details see the article by (US) Lt Al Johnson, GPS Y2K Lead Engineer, MILLENNIUM (Y2K) AND GPS END OF WEEK (EOW) ROLLOVER Major suppliers of GPS navigation devices claim to have prepared their computational systems for this day. However, incidents on the actual GPS EOW roll-over day are possible.

Which is the Millennium Satellite ?

The ARIES Satellite will be launched in 2000. FedSat 1 will be launched in 2001. Which of these two Australian satellites is the Millennium Satellite?
Australian Prime Minister John Howard has declared that the next Millennium starts in

From: Ed Yardeni <yardeni@ix.netcom.com>
To: econews-recipients@x86.webcom.com <econews-recipients@x86.webcom.com>
Date: Monday, April 13, 1998 9:41 PM
Subject: Dr Ed's conference call with ex-CIO of IRS

Arthur Gross recently resigned from his position as the Chief Information Officer of the IRS. He was the special guest on my Monday morning conference call. We discussed whether IRS computers would be ready for the year 2000. Mr. Gross was relatively optimistic about the readiness of the agency's software. However, he is very concerned about the telecommunications systems.

To hear the rebroadcast, call 1-800-633-8284 access code 3241594.

Dr Ed

CONFLICTING STORIES

April 13, 1998

IRS: Rescue

Jeffrey H. Birnbaum

To the rescue rode Art Gross, 53, a veteran of the New York State revenue department. Gross, the new chief information officer, said publicly that the IRS's computers didn't "work in the real world" and that its employees lacked the "intellectual capital" to transform them. In 1996, when Gross arrived, the IRS's year-2000 conversion project had a budget of $20 million and a staff of three; now it's a $900 million project with 600 workers, many of them consultants. There's at least a chance the whole tax system won't shut down on Jan. 1, 2000, which was distinctly possible before Gross' arrival.

IRS: Unbelievable!

Gross' major challenge was to try, once again, to fix the big machines. First he did triage. He popped the Bubble Machine and a couple of dozen other wayward projects, including a hastily drafted program to allow taxpayers to file over the Internet. Then, with help from TRW, he devised a new top-to-bottom computer architecture. At its heart was a centralized database that would make the IRS more like American Express and less like the Pony Express. Gross wanted to give to any taxpayer who called a status report on his or her account, something all but impossible now. He also was determined, at last, to start entering tax return information electronically.

What if a Post-it falls off? These "fats" are too large to be sorted by machine.
Photo: Adam Bartos

But then it all went wrong again. The Treasury Department, in its zeal to overhaul the most broken arm of government, brought in a new IRS commissioner--Rossotti--who was an expert in the same field as Gross. A clash was inevitable. As far-reaching as Gross' plans were, Rossotti's were even more ambitious. On top of year-2000 changes, computer updates from the 1997 tax law, and the overall modernization, Rossotti proposed to restructure the entire organization. The combination was too much for Gross. With evident exasperation, he says, "I chose to move on with my life." To conceal Gross' discontent, a Clinton official whispered that his decision was purely personal, that he had a girlfriend in another city, and that he wanted to be closer to her. Gross' significant other in fact lives in Florida. But his new job, at a medical center in Albany, N.Y., is 300 miles farther from her.

Gross told FORTUNE that he was proud of the modernizations he championed. But there is reason to worry that the job he started might never be completed. Gross believes the creation of a centralized database "requires a situation in which the CIO [chief information officer] is invested with all the authority and resources to make it happen." And that, he says, "is not a certainty." Gross and others also argue that by planning to restructure later, Rossotti is putting the proverbial cart before the horse. Currently the IRS is organized geographically into 33 districts that report to ten regional offices. Rossotti contemplates a radical change to a system organized by business units. The IRS would have separate divisions for individual taxpayers, big businesses, tax-exempt entities, and small businesses. Experts question the wisdom of beginning a technological conversion before the organization is set. "An effective information-systems architecture is designed around the business it supports, not the other way around," says Jim Woldbarsht, ex-CIO of the Pension Benefit Guaranty Corp., now a software executive. For his part, Rossotti says the computer blueprint as it stands can be adapted to his plans.

Many unhappy returns: CIO Gross quit the IRS in frustration.

Photo: Adam Bartos

Gross' other doubts stem from the way Washington works. "This is a great town for policy," he says. "Policy entails wonderful debate and negotiation and brokering and all that. But once you commit to a technology direction, the key to success is very aggressive, unremitting management, without a lot of room for ongoing debate. In the two years I've been here, it appears that this town is not always conducive to that kind of management."

Gross and other observers say the relatively low government pay scale (he earned $125,900) could prevent Rossotti from bringing in the talent he needs. "There are enormous opportunities in the [informational technology] industry to make large personal fortunes," Gross says. "It's an exception when you can obtain very senior technology-management people to commit to this kind of effort in government." Gross sees himself as that kind of exception, and is clearly leaving with regrets. "I'm a finisher," he says wistfully, knowing well that he hasn't finished much.

Rossotti is more upbeat. He is hopeful that Congress will let him hire eight or more managers at an annual income of $172,000, including bonuses. That ought to be enough to lure capable employees. But the IRS needs more than a dozen good men and women. Even under the best of circumstances, modernizing its computers could take as long as 15 years. Longtime IRS employees have come to view computer plans with skepticism. In Philadelphia, section chief Robert Davis describes one mainframe as "ancient" and another as a "hodgepodge" and shakes his head when he asserts, "Everything you see is obsolete and is going to be replaced." But then he adds, "*If* everything goes well--I always have to add the 'if.' " At the IRS they've heard it all before.

So what happens if nothing happens? What happens if the machines can't be fixed? One possibility is systemwide collapse due to sheer obsolescence. Another more likely outcome is a replay, perhaps on a bigger scale, of the 1985 Philadelphia meltdown. Unfortunately, that's probably what it will take for Washington to drag the IRS into the modern age.

Back

Next

The IRS is taxed by Y2K

, PC Week
April 3, 1998 2:59 PM PST

As millions of Americans prepare to file tax returns before April 15, the Internal Revenue Service is facing a deadline of its own: Sept. 30, 1999, the day the agency wants to complete testing of its Y2K conversion and give a thumbs-up for the "00" date roll.

Having systems unprepared for the year 2000 date change has daunting repercussions of possible downtime or revenue losses for most of corporate America. However, the impact for the IRS could be far more devastating, given that every American could be affected if the agency's computers can't collect taxes or issue reimbursement checks.

Acutely aware of its responsibility to the public and the government--President Clinton recently issued an executive order mandating compliance within every federal agency--the IRS is taking this job very seriously. Commanding the effort is John Yost, century date change project director at the IRS, in Washington. "In terms of individual components, we are about 55 percent of the way through the implementation," says Yost.

That implementation comprises 80 IBM and Unisys Corp. mainframes, as well as 1,400 minicomputers, including models from Sun Microsystems Inc., Sequent Computer Systems Inc., Pyramid Technology Corp. and Hewlett-Packard Co. In addition, the IRS has another 130,000 desktop and network devices that must be checked for compliance.

A herculean project in its own right, the conversion process has recently become further complicated by internal changes and outside influences. On April 1, Arthur Gross, CIO of the IRS and the primary Y2K advocate in the agency, resigned from his post. No replacement has been named--which means no Y2K bellwether. This may be dampening the spirits of the Y2K team, which is reportedly looking toward opportunities in the private sector, where the pay is better.

"Because there is huge competition for programmers, the attrition rates have more than doubled over the last two years," admits Yost.

In an attempt to neutralize the defection rate, IRS officials are offering 1,000 employees and contractors working on Y2K conversion a 10 percent retention allowance this year and possibly next year as well.

In addition, the IRS is promising programmers long-term job security and is holding out the carrot of a $3 million increase in this year's IS training budget.

Overall, the IRS expects to spend just under $1 billion to get ready for the date change. While the pressure is on, this is

one project that does not get muddy in political waters. "I've never worked on a project before where funding is as readily available," Yost says.

This type of high-level support, says Yost, should yield good results for the IRS. So don't plan on missing the April 15 deadline in 2000.

Y2K Stats: Internal Revenue Service

EMPLOYEES: 700

CONTRACTORS: 300

PROGRAMS TO TRANSLATE: 85,000

LINES OF CODE: 50 million

SPECIAL ISSUES: As a federal agency, the IRS must comply with an executive order issued in February by President Clinton regarding Y2K conversion. Under the policy, agencies must offer assurance that no critical federal program experiences disruptions because of the Y2K problem. However, the IRS infrastructure is spread across 800 sites throughout the country, which makes upgrading and testing everything an unruly process. Internal changes at the organization, including the resignation of IRS CIO Arthur Gross on April 1, are adding to the complexity of the problem along with the fierce competition for programmers, who are tempted away from the IRS by larger salaries offered in the private sector.

DEADLINE FOR COMPLETION: September 1999

COST: $1 billion

Deseret News *Utah news*

Bennett to head panel on Year 2000 problem

Senate aims to head off costly computer crashes

Last updated 04/04/1998, 12:01 a.m. MT

By Steve Fidel
Deseret News staff writer

A Senate committee created Thursday to address the computer-crippling Year 2000 technology problem will be led by Sen. Bob Bennett, R-Utah, the senator announced Friday.

The problem is simple: Two-digit date codes in computer hardware and software will leave machines not knowing which century it is when the calendar turns from 1999 to 2000. Computers may crash or start spitting out error codes if they can't tell whether incoming data is referring to 2000 or 1900.

Problems are already cropping up, Bennett said. Machines that process credit card purchases stumble on cards with expiration dates in 2000.

Bennett has had a keen interest in the Year 2000 or "Y2K" problem for some time. He already has sponsored legislation requiring publicly traded companies to disclose their Y2K compliance status.

Fixing the problem is simple, Bennett said: Devices with deficient calendar-sensitive hardware need that hardware replaced; software with calendar-sensitive programming codes need the codes rewritten.

Finding the problem is where the trouble is. Software that needs fixing is scattered across an aggregate of 100 billion lines of computer code, and computer chips that are likely to fail are imbedded in a countless number of machines from wristwatches to medical devices to government supercomputers.

Bennett estimated the price tag for the fix at $1.4 billion for the Treasury Department alone, not including the Internal Revenue Service; $10 billion for all of the computers in the United States; and $600 billion for computers worldwide.

The new Senate committee will have seven members, four Republicans and three Democrats, who have yet to be named. Bennett said the committee will also have adequate technical staff, "but major appropriations for fixes need to be made."

"The committee will provide oversight and legislative recommendations to help the government and private sector react more quickly and effectively to the economic difficulties arising from the Year 2000 system failures," Bennett said.

Forbes ASAP

QUICK SEARCH: Contents 04.06.98

COMPUTER MELTDOWN

Bullish economist **EDWARD YARDENI** offers a pessimistic prediction about the Year 2000 Problem.

Edward Yardeni, chief economist at Deutsche Morgan Grenfell, has long touted the importance of technology. But lately Yardeni is sounding a bit bearish. He worries that the Year 2000 Problem may have a bigger impact on the economy than many people suspect.

ASAP: There has been a lot of hype about the Year 2000 Problem. Do you think that some consulting firms are overhyping it to bring in more business?
Yardeni: Some may be, but most of them have plenty of other information technology business. They don't have to hype the problem.
ASAP: Then if it's not overhyped, why is the Year 2000 Problem so important?
Yardeni: There is no scientific, well-documented way to write code. Writing code is very creative and casual. As a result, going back to fix the code is going to be difficult. It will be a case-by-case process. There are no easy solutions.

And there is no way we're going to fix 100% of all the computer systems around the world in time. There are going to be disruptions.

If the disruptions are significant, we could have a very nasty recession. My analogy is the 1973-74 recession. Just the way a disruption in the supply of oil caused a global recession, a disruption in the flow of information, especially if it is critically important information, might similarly disrupt global economic activity and produce a recession.
ASAP: What kind of specific disruptions could we have?
Yardeni: I've read that the Federal Aviation Administration is having a lot of problems getting its computers ready for the Year 2000. The chief information office of the IRS has made it clear that he's very concerned that it is going to be a close call whether the IRS computers are functioning in the year 2000. A lot of people may want to see the IRS computers melt down, but that's the way the government collects revenues. Revenues are necessary for all the spending that the government does. Everybody is a beneficiary of that, one way or another.

COST OF FIXING Y2K

EXPENSES FROM 1994-1999	IN BILLIONS
Initial software repairs	$ 530
Secondary "bad fix" software repairs	$ 50
Test library repairs	$ 75
Database repairs	$ 454
Hardware chip replacements	$ 76
Hardware performance upgrades	$ 150
Subtotal	**$1,335**
EXPENSES FROM 2000-2005	
Litigation and damages	$ 300
Post-2000 damages	$ 580
Post-2000 recovery expenses	$1,406
Subtotal	$2,286
Total	**$3,621**

Source: Software Productivity Research

But my main concern is the supply of electricity. The utilities are reacting to the problem. However, they are vulnerable to third-party risks. For example, the NRC [U.S. Nuclear Regulatory Commission] might have to shut some reactors down if all their systems are not ready. Fossil fuel burning plants must get fuel deliveries during the winter of 1999-2000. This could be a problem if the railroad system is disrupted by the Year 2000 problem. Electricity is my main concern, not so much because I believe it is the most vulnerable system, but because without the juice we won't be able to fix all the other systems.
ASAP: What about business? Are large corporations prepared?
Yardeni: Based on survey data I have seen, they will be well prepared. The only problem is that they are very dependent on hundreds of thousands of customers and vendors who may not be ready. So the big companies' operations might be seriously disrupted by the failure of smaller firms. They would also suffer if the government is unable to function properly.
ASAP: Are most people complacent because they think technology will save itself?
Yardeni: Absolutely. People think somebody will come up with a simple solution that you can spray all over your computer on Friday, and it will be working perfectly fine when you come into work on January 2, 2000. We're kidding ourselves if we think we can come up with a contingency plan at the last minute.

—Interview by Geoff Baum

Back up

News

Headlines
Top News Summary

- Paper: Starr Report Under Way
- Grand Jury Testimony Resumes
- NTSB Seeks 747 Wiring Checks
- Clinton to Press Programs
- Ford Going With Side Air Bags
- Optimism Over N.Ireland
- Study: 1 in 8 Plants Threatened
- Bono's Widow Elected to Congress
- Americans Need More Vitamin B
- Russian Showdown Brewing

Personalize **Headline** news
seek

Choose another news channel:
Headlines go

Personalized news

Personalized news:
Computer news

Personalize all news categories

Full story

Global Alliance Called for to Fix Year-2000 Bugs, Avoid Recession Series: 6

American Banker
Tue, Apr 08 1998

International bank regulators were warned Tuesday that year-2000 computer problems will cause a global recession unless extraordinary steps are taken now.

The industrialized nations must create a global, military-style alliance to contain the computer problem and appoint General Colin Powell or someone of similar ability as its commander in chief, said Edward Yardeni, chief economist for investment bank Deutsche Morgan Grenfell.

"Let's stop pretending that Y2K isn't a major threat to our way of life," he said in a speech to the Bank for International Settlements in Basel, Switzerland. "The current Y2K global battle plan is virtually guaranteed to fail."

Mr. Yardeni's recommendations were part of a seven-part plan that includes a freeze on regulatory and statutory changes affecting information technology and a mandatory holiday during the first week of January 2000.

Of particular interest to financial institutions, Mr. Yardeni recommended the creation of "industry alliances" that would help assure key activities, such as banking services, would not be disrupted.

"As representatives of the world's banking and financial

the coming upheaval," he told Bank for International Settlements members.

Mr. Yardeni, who gained attention earlier this year by winning a Wall Street Journal contest on economic forecasting, began assessing the impact of the year-2000 problem last July. At that point, he gave the year-2000 problem a 30% chance of causing a global recession and has since raised the estimate steadily.

In his speech Tuesday, Mr. Yardeni said the likelihood of recession would be "closer to 100%" if the G-8 nations do not act at their May 15-17 meeting in Birmingham, England.

Even if the G-8 does act, he said there is a better than even chance of a recession as severe as the 1973-74 one, when the U.S. gross domestic product dropped 3.7% from peak to trough. He said the U.S. nominal GDP and stock market capitalization could both fall $1 trillion.

"Information is just as vital as oil for running our economies," he told the Bank for International Settlements conference, which was also attended by members of the Committee on Payment and Settlement Systems, the International Association of Insurance Supervisors, and the International Organization of Securities Commissioners.

In an interview before the speech, Mr. Yardeni backed off of an earlier prediction that 20% of small financial institutions would fail due to the year-2000 bug, saying it was an off-the-cuff "guesstimate" with no basis in econometrics.

But he did say the 20% failure rate is a plausible worst-case scenario. "If there are going to be business failures, there are going to be nonperforming loans."
http://www.americanbanker.com

WESTERGAARD year 2000
strategic analysis of the y2k problem

Tuesday,
April 7 1998

the MENU

economy
industry
international
investing
litigation
media
technology
washington
home

search this site
what is y2k?
mission statement
contact us

other Westergaard sites:

Westergaard Online

BioMedtech Online

Westergaard Broadcasting Network

April 7, 1998

Y2K Issues and Recent Developments in Congress

By Richard Nunno

Based on the frequency of hearings, and other congressional activity, I am of the opinion that congressional interest in Y2K is intense and, if anything, increasing. On March 18, The House Government Reform and Oversight Committee, Subcommittee on Government Management, Information and Technology, and the House Committee on Science, Subcommittee on Technology, held another joint hearing, as part of their ongoing oversight of federal government's progress on fixing the Y2K problem for their systems. Witnesses in the hearing's first panel included (in his first appearance before Congress) Mr. John Koskinen, the recently appointed Chairman of the President's Council on Y2K Conversion, Mr. Gene Dodaro, Assistant Comptroller General of the U.S. General Accounting Office (GAO), and Mr. Michael Harden of Century Technology Services Inc., a Y2K research firm. Mr. Koskinen gave a general overview of his conception of his role to promote the efforts already underway at federal agencies and to work with state and local governments, the private sector, and foreign and international organizations. Mr. Dodaro emphasized that federal agencies have made insufficient progress, and pointed out that "at the current pace, it is clear that not all mission critical systems will be fixed in time." Mr. Dodaro further stated that "risks of disruption to government services is high," and that "key economic sectors are at risk of Y2K failures."

For the second panel of the hearing, the joint congressional committees reviewed Y2K progress in the Department of Treasury. Witnesses included the Treasury's Deputy Assistant Inspector General, the Chief Information Officers of both the Treasury and the Internal Revenue Service, and the Commissioner of Information Resources of Financial Management Services. Although the Treasury Department officials emphasized that significant progress had been made on Y2K conversion, subcommittee members raised questions regarding their ability to meet the January 1, 2000 deadline for all mission critical systems.

weekly breakdown

MONDAY
Y2K Tip of the Week
Y2K Radio Rap

TUESDAY
Washington Interpreter
Y2K On The Job

WEDNESDAY
Technology Corner
Digital States & Associations

THURSDAY
Investment Insights
In The Trenches

FRIDAY
International Watch
"POWER"-ful Prognostications

As part of the Committee Members opening remarks, Representative Barcia, the new ranking Member on the Technology Subcommittee, mentioned that the Y2K preparedness of some legislative agencies might also be at risk. Barcia stated that two reports by the House Inspector General (one from December 1996 and the other from December 1997) indicated that the House Information Resources had not resolved its Y2K issues. Barcia asked that the two reports be placed in the record.

Also on March 18, the Senate Banking Committee, Subcommittee on Financial Services and Technology held a hearing to investigate the efforts of the federal Office of Thrift Supervision (OTS) to ensure that the nation's thrift savings and loan institutions are Y2K compliant. The director of OTS assured the Subcommittee that it was working diligently to correct its own systems and was examining the Y2K conversion plans of all 1,200 thrifts across the country. A representative from GAO, however, testified that many thrift institutions are not making sufficient progress, and risk system failures that "could lead to thrift closings and serious disruptions to both the thrift community and customers." GAO noted that OTS's problems were similar to those that GAO had previously reported for the Federal Deposit Insurance Corporation and the National Credit Union Administration which were mainly starting late and having limited resources.

In the evening of the 18th, Senator Bob Bennett, Chairman of the Senate Subcommittee on Financial Services and Technology, spoke at the Washington DC Y2K policy forum sponsored by Fannie Mae, the federally chartered, housing loan agency. The Senator discussed his communications with FED Chairman Alan Greenspan on the Y2K issue, and was assured that there will be enough cash available in the event of a liquidity crisis due to Y2K problems. In response to a question from a computer programmer, Bennett said that he would consider Y2K safe harbor legislation that would provide liability protection to companies and computer programmers. Bennett also mentioned that Senators Lott (majority leader) and Dascle (minority leader) were considering forming a new Senate Committee that would have jurisdiction across all agencies and industries with the single task of looking into the Y2K Problem. Bennett indicated that he was hoping to be appointed chairman of the new committee. To date, no announcement of the formation of a new Senate Y2K committee has been made by the majority or minority leader's offices.

On March 20, President Clinton signed into law H.R. 3116, the Examination Parity and Year 2000 Readiness for Financial Institutions Act (becoming Public Law 105-164). This legislation gives the Federal Office of Thrift Supervision and the National Credit Union Administration authority, similar to that of other banking regulatory agencies, to examine the operations of contractors that perform services for thrifts and credit unions, such as data processing and the maintenance of computer systems. The Act is designed to assist federal regulators in better understanding the Year

2000 risks faced by financial institutions in order to more effectively oversee their operations. The original bill, introduced on January 28 by Representative Jim Leach, chairman of the House Banking Committee, had moved through Congress swiftly, passing the House by over two-thirds majority, and passing the Senate by unanimous consent.

On March 24, the House Committee on Banking and Financial Services held a hearing to assess the Y2K readiness of critical systems at the Departments of Housing and Urban Development, Treasury, and the Federal financial regulatory agencies. The new so-called Y2K "czar" John Koskinen testified for his second time before Congress, along with two GAO officials. Again, Members and the GAO witnesses raised serious questions about the Y2K preparedness of these agencies and their ability to meet the deadline.

On March 30, COMDEF '98, an international defense conference, was held at the National Press Club in DC, at which the first issue presented was the Y2K Problem--national security implications. Mr. Frank Gafney, Director, Center for Security Policy, spoke to the group about the potential problems that the U.S. government and its allies might face if the problem is not resolved in time. After criticizing the Administration's policies on Y2K, Mr. Gaffney stated that Y2K could leave this country vulnerable to a limited information warfare attack, and that he believed that we need a "Manhattan Project" style emergency program to investigate and select specific policies and programs that can accelerate the progress in government agencies and the private sector. Gaffney described his idea for a government-sponsored, visible, highly publicized competition among computer companies to develop a technical solution to Y2K to spur industry on toward bringing more advanced software tools to the market.

Finally, on April 1 the Senate Governmental Affairs Committee held a Y2K hearing with Mr. Koskinen, along with the Deputy Secretaries of the Departments of Health and Human Services and Transportation. Titled "Crashing into the Millennium," the hearing investigated many questions regarding federal agency Y2K conversion efforts. Some of the issues raised include the following:

1. Ensuring that the interfaces between state, local, and federal computer systems remain operational,
2. Helping small businesses and consumers cope with the year 2000 problem. While larger companies might be able to prepare their systems for the date conversion, small firms might not have the resources or the time to meet the deadline,
3. Determining the status of foreign countries and international organizations with regard to fixing their computer systems, and assessing the potential impact on U.S. systems,
4. Determining whether U.S. air traffic control systems will inter-operate with foreign systems so that international travel will be safe after January 1, 2000,