Business
Substances
Environment

Information
for the
Underwriter

apropos The millennium bug

Contents

Introduction
The problem
Origins of the problem
Systems affected
- Mainframes
- Personal computers
- Embedded chips/systems

Scale of the problem
- Costs
- Dates and programming
- Time
- Resources
- Interdependencies

Solution to the problem (project management)
- Initial phase
- Stocktaking phase
- Planning phase (analysis and security)
- Implementation and test phase

Tips for the underwriter

1.20

AssTech
Sederanger 4 – 6
D-80538 München
Germany

Introduction

In a global economy that is increasingly dominated by information technology there is a growing dependence in all areas on computer systems.

Malfunctions or breakdowns of hardware or software can cause companies considerable damage and financial losses, which, taken together, are not without consequence for whole national economies.

The start of the new millennium marks a watershed in information technology: the change in the date to 2000 will pose a very complex and difficult-to-solve problem for all companies and government bodies who use computer technology, no matter what type or on what scale.

The problem is that, in computer programs, years in dates were expressed in two-digit and not four-digit format. The difficulties posed by the two-digit format have in fact always been present in latent form and have been recognised ever since modern information technology has been in use. However, the problem has only relatively recently started receiving media coverage, while companies are just beginning to come up with concrete solutions and to implement these in projects.

At present no one has any real idea of how big the date problem is going to be, but IT and computer experts expect it to have a whole range of consequences. At worst, systems around the globe will crash, posing a serious threat to the world economy. At the other end of the spectrum, some believe the problem will be no worse than any other of the computer problems that people have to deal with every day.

The problem

The "millennium bug", as the media calls it, but also known as the "time bomb" or "doomsday", is due to the fact that most computer programs written in recent decades use a two-digit format to represent the calendar year in dates (e.g. "97" instead of "1997"). The four-digit format showing the current century was considered superfluous.

In the programs under threat from the millennium bug, the date format is six-digit, e.g. 31 October 1997 is represented as 31/10/97 in Europe and as 10/31/97 in the USA. At the start of the new millennium, the year 2000 will be shown as "00" in programs having a two-digit year field. As the computer treats every date as being in the 20th century, it cannot differentiate between the years "2000" and "1900" and will interpret "00" as being the year "1900". This can lead to errors in calculations that depend on the date.

Further problems arise with the comparison of dates, if such a comparison is based on the assumption that the numeric value of a date increases with time. Thus, for example, the date 1 August 1997 (= 970801) has a greater numeric value than 5 September 1996 (= 960905). This assumption is no longer valid in programs with a two-digit year format after the turn of the millennium: in this case, 1 January 2000 (= 000101) has a lower numeric value than 8 September 1998 (= 980908).

The following three cases exemplify the problem. They show simple operations using two-digit dates on the one hand and four-digit dates on the other.

Operation	four-digit year field	two-digit year field
Calculating age:		
Year of birth 1962		
Age in 1999	1999–1962 = 37	99 – 62 = 37
Age in 2000	2000–1962 = 38	00 – 62 = –62
Calculating a deadline:		
Use-by date		
1. August 2000		
Current date	20000801 > 19971201	000801 < 971201
1. December 1997	product still good	product no longer good
Sorting by year:		
1964, 1991, 2000, 2011	1964 –1991– 2000 – 2011	00–01– 64 –91

Computer operations based on a year field "00" can not only deliver incorrect results and destroy data, but also cause programs and whole systems to crash.

However, the problems posed by two-digit dates will begin well before the new millennium, i.e. before 1 January 2000. Many forward calculations (e.g. long-term calculation of interest or setting a use-by date) will need to access critical dates well before the new millennium, which is why problems have already arisen with loan or insurance contracts where terms extend beyond the turn of the millennium.

The millennium problem is further aggravated by the fact that the year 2000 is also a leap year. In the past, leap years have often led to considerable problems in IT systems that have not been correctly programmed.

In a New Zealand smelting plant, for example, millions of dollars worth of damage was caused on 31 December 1996 as a result of overheating. The ultimate cause of the incident was faulty software, which failed to recognise that 1996 was a leap year and promptly shut down the control computer without warning.

Origins of the problem

In the early 1960s, when companies first began installing computers, the two-digit year field was generally accepted programming practice, for which there were both technical and economic reasons.

From the technical point of view, the memory space available in systems for programming and data storage was usually quite limited. In view of the fact that system resources were limited and expensive, programmers did their best to avoid allocating storage space unless absolutely necessary. One method of saving storage space was to omit the constant "19" at the beginning of each year. As this approach helped keep data stocks low – thus also reducing costs – it was decided to omit these digits.

Another argument in favour of the two-digit year presented by software developers at the time was that the systems in question would no longer be in use at the turn of the millennium, having already been replaced by year-2000-compliant ones. But, in fact, many of the systems developed at that time have continued to grow and are still being used today. What is more, the two-digit year field remained accepted programming practice until very recently, even though it was no longer either necessary or advantageous owing to the much lower costs of memory space.

Systems affected

The millennium bug can occur world-wide and will affect hardware and software equally. The types of systems at risk range from the mainframes used by international providers of financial services to the ubiquitous PCs, and the chips in domestic video recorders.

Mainframes

It is likely that the highly complex, and often rather old, mainframe programs will suffer relatively serious effects. Over many years these programs have grown, been fed with information and adapted to meet the changing structures of the companies in which they are installed. With systems like these, the changeover to the new millennium will require considerable time and effort. This is further complicated by the fact that they are based on programming languages such as FORTRAN and COBOL, which are no longer in common use.

That many companies handle a large number of essential business processes on systems of precisely this kind, and that the number of programs used in this context is very large, only serve to exacerbate the problem in this segment.

Personal computers

In the case of PCs, the problems may be located in either the fully integrated hardware or the system or application software.
The date problem can affect the following components of the PC:

- system clock (CMOS real-time clock)
- BIOS ROM (basic input output system), controls operations of the PC components
- operating system (DOS, Windows, OS2, etc.)
- application software

As the individual elements rely on each other for the date, all components have to be compliant with the year 2000 so that the whole PC system can function.

Embedded chips/systems

Embedded chips and systems are electronic components that perform control and measurement functions and that are integrated in industrial facilities as well as in a wide variety of devices used every day. At the simplest level this can be an individual microprocessor (chip) carrying out only a single function, e.g. processing the data of a temperature or gas sensor. In complex systems a large number of embedded chips often work in combination, under the control of specialised software applications.

The following are some of the areas and corresponding systems in which embedded chips are usually integrated:

Building technology:
- lifts, escalators
- fire and other alarm systems, sprinklers
- surveillance cameras, access control systems
- heating and air-conditioning systems
- safes and strongrooms

Offices:
- telephone answering machines, photocopiers, fax machines
- time data entry systems (e.g. to record working hours)
- telephone systems

Production/workflow control:
- manufacturing facilities and production lines
- control systems, e.g. chemical plants and refineries
- industrial robots
- waste (water) management systems

Transport:
- air-traffic control systems
- traffic lights
- ships, aircraft, trains, motor cars

To ensure that embedded chips are year-2000-compliant, it is necessary to identify all the components that have been programmed with date logic. In contrast to software systems, fixed-code chips cannot be reprogrammed as it is generally impossible to access the integrated code in order to change it. The components that depend on date information can be identified by consulting the accompanying documentation or by contacting the manufacturer. For systems that are not year-2000-compliant, the chips or units in question have to be replaced completely.

Scale of the problem

The millennium bug is easily the greatest challenge the information technology industry has had to face in its brief history. It is a problem that will occur simultaneously and to more or less the same degree in the affected companies and organisations across the globe. The underlying cause is easy to describe and, from the point of view of programming, seems almost trivial. However, the real difficulty, and what requires so much effort, is the huge number of affected program parts and data records and the fact that they are to be found in all areas of a company.

Costs

According to often quoted estimates of the Gartner Group, the total cost of altering the date in the affected software is likely to be around US$ 300-600 billion. This figure is based on the assumption that some 250 billion lines of program code are affected by the problem and that, on average, it would cost between $ 1.10- and $ 1.70 to analyse, adapt and test each of these. Apart from the outlay for modifying the software, the estimate also includes the replacement of computer chips that are not year-2000-compliant. In the USA alone, the cost of potential lawsuits in connection with the millennium bug is put at $ 1,000 billion; the first such lawsuits are already pending.

For Germany and the UK the cost of conversion is estimated to be DM 50 billion and DM 100 billion respectively.

The potential damage to the economy if the systems are not converted in time would probably amount to a multiple of the cost of conversion.

Data and programming

What makes solving the millennium bug so difficult is the huge amount of programs, code lines and data that has to be checked and possibly adapted. A standard medium-sized company uses an average of several thousand mainframe programs for its business processes; these, in turn, consist of several million lines of code. The challenge lies in identifying and adapting the large number of fields and variables with a date reference. As many programs are connected with each other in a huge variety of different ways, it is necessary to analyse the exact program structures before any adaptation takes place. This is made more difficult by the fact that many of the older programs are documented only poorly or not at all, and the developers of these systems often no longer work for the companies in question.

Further hurdles to be negotiated on the road to year-2000-compliance result from the need to adapt and reprogram the systems themselves. Experience shows us that corrections and maintenance work inevitably lead to new errors in the programs, errors which are often not immediately recognisable and which affect other system components.

Time

The time frame available for solving the millennium bug is a finite one, ending at the latest on 31 December 1999. All companies and organisations world-wide are tied to this deadline. Given the dimensions of the problem, the time remaining until the end of the century is relatively short, and the risks increase with each unproductive day that goes by.

This is especially true when one considers that many ordinary IT projects are not completed by their planned deadlines. So as to have enough time to test the adapted software, it is advisable to have completed conversion during 1998. Testing of the adapted systems will take a considerable amount of time, between 40 % and 60 % of the total project time according to estimates.

Resources

Another problem is the personnel resources available for adapting the systems. In many countries there is already a shortage of qualified IT personnel, and this will be further exacerbated by the millennium bug. Many companies are currently involved in setting up their own intranets and, in Europe, managing the changeover to the Euro, another time-consuming task. The latter has to be accomplished simultaneously with solving the millennium bug, thus utilising more of the already scarce IT resources. As many of the programs affected by the millennium bug are written in programming languages no longer in current use, the search for qualified personnel is particularly difficult. In the USA, for example, the hourly rates for COBOL programmers doubled in 1996. According to current estimates, the available personnel resources will be insufficient to meet the expected demand from companies.

Interdependencies

Even if the individual company or organisation ensures its year-2000-compliance in time, this is no guarantee that it has solved the problem completely. This is because a company's interfaces with its customers and suppliers also constitute sensitive areas in this regard. When computer data is exchanged, data from systems that are not year-2000-compliant can cause problems in those that are. One example is the networking of production plants or department stores with their many suppliers. This is why projects dealing with the millennium bug should not concentrate simply on compliance within the company itself, but also consider that of its business partners.

Solution to the problem (project management)

In view of the complexity of the problem, the scarcity of the resources required and the tight deadline to be met, solving the millennium bug places great demands on project management. For, in contrast with most other software projects, the deadline cannot be extended. Given the limited resources available, it is essential to set priorities. In many companies, a mere 20 % of business processes account for as much as 80 % of turnover. It is thus necessary to identify those processes that are essential to maintaining a company's business viability and to ensure that these processes at the very least are made year-2000-compliant.

The following is a sample year-2000 project plan, showing the various project phases and the corresponding tasks involved.

Initial phase:
Owing to the high degree of networking and the large number of applications involved, the millennium bug project is an interdisciplinary one affecting a variety of company divisions and departments.

- Heightening awareness of the problem within the organisation
- Informing and integrating management in the project
- Setting up a millennium-bug project group comprising staff from all affected areas
- Interviewing and informing staff

Stocktaking phase:
In order to create a basis for planning, it is necessary to gauge the scale of the problem throughout the company.

- Identifying business-critical processes and how these interrelate with IT systems
- Drawing up a complete software and hardware inventory (including embedded systems/chips)
- Documenting the company's external/internal interfaces
- Carrying out pilot projects to ensure a realistic basis for planning

Planning phase (analysis and strategy):
Concrete procedures are determined based on the data gathered.

- Estimating the potential time and costs
- Working out a schedule and allocating the personnel resources (including any outside service providers)
- Determining priorities and work packages
- Deciding on the methods and tools to be used (e.g. support tools for analysis and conversion)
- Developing a test strategy
- Possible replacement of the company's own programs with standard software

Implementation and test phase:
The individual programs are reprogrammed or replaced. Subsequently, not only the modified programs, but the company's entire IT system, have to be thoroughly tested.

- Adaptation and implementation of each application's source code and data
- Program tests of the modified applications
- Carrying out of integration tests
- Company-wide system test for year-2000-compliance including, where possible, suppliers and customers

Companies should see the modification of IT systems necessitated by the millennium bug not just as a problem, but also as an opportunity. It offers a chance to analyse, review and modernise existing systems.

Tips for the underwriter

The millennium bug may have consequences for the following classes of insurance:

Product liability insurance

In this class, it is above all manufacturers of hardware and software or producers of computerised control systems that are more highly exposed. As a rule, under product liability cover a company manufacturing products for end users may be faced with claims for damages if those products are defective.

Possible case: a company manufactures a computerised production facility that is not year-2000-compliant and ships it to a customer. Owing to the millennium bug in its software, the facility then causes bodily injury and property damage.

Professional indemnity insurance

Here the main liability problem is posed by IT and software consultants who provide their clients with incorrect or insufficient information with respect to the millennium bug. External auditors, too, may be held liable if they neglect to take the millennium bug into account when carrying out their job.

Possible case: an IT consultant advises a company on the purchase of a new computer system. As it is not year-2000-compliant, the system generates incorrect lists of outgoing merchandise. The result is financial loss to the company.

D&O insurance

Almost irrespective of the branch of industry involved, D&O covers are highly exposed to the problem of the millennium bug. This is because damage to third parties arising from management's incorrect assessment or denial of the millennium bug are conceivable in all companies.

Possible case: a company's management considers the millennium bug to be of minor importance and cuts the budget of the relevant project group. In the year 2000, a number of essential IT systems fail, resulting in a dramatic drop in profits for the company. As a result, the shareholders sue the company's board of management.

In the third-party liability field, commercial TPL and environmental liability (e.g. incorrect control of environmentally relevant facilities) are other areas at risk from the millennium bug. However, the property insurance field, too, may be affected by the millennium bug, e.g. owing to business interruptions or property damage.

The insurer has several options for minimising the risk to his portfolio posed by the millennium bug. A few of these are listed below:

- Sensitise customers and heighten their awareness of the problem (information material and discussions)

- Co-operate with insureds to carry out a risk analysis in order to gauge their exposure to the millennium bug

- Set down clients' exposure in writing (obligation to disclose year-2000 problem in the policy application) and treat information given in application as a warranty

- Draw up and use a special millennium-bug questionnaire to help assess potential risks

- Advise the insured in loss prevention and project management

- Insist on adherence to specific requirements and conditions with regard to the millennium bug

- Introduce or utilise specific exclusions for highly exposed risks

All in all, the problems associated with the millennium bug will make underwriting more difficult and time-consuming in the classes of business involved. Some insurers are already offering new insurance policies that are specially designed to cover the year-2000 problem.

Postal address.
AssTech
Assekuranz und Technik
Risk Management Service GmbH
Postfach 22 11 50
D-80526 Munchen
Germany

Office address:
AssTech
Assekuranz und Technik
Risk Management Service GmbH
Sederanger 4 – 6
D-80538 München
Germany

Telephone: +49 89 3844-1585
Telefax +49 89 3844-1586
Telex: 52 152 47 bav d

Spanish subsidiary:

BEER & AssTech, S. A.
RISK MANAGEMENT SERVICE, S. A.
Miniparque Empresarial
El Soto de la Moraleja
Azalea, 1 – 3 (Edificio E)
E-28100 Madrid

Telephone: +34 91 650 91 42
Telefax: +34 91 650 95 14