|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | |
|  | Y2k Bug | | | | | |  |
|  | | | | | | | | |
|  | | | | | | | | |
| **How we planted our time bomb:**  **In the 1950s and '60s, representing years with two digits became the norm. One reason for this was to save space. The earliest computers had small storage capacities, and only a fraction of the**[**RAM**](https://en.wikipedia.org/wiki/Random-access_memory)**of modern machines. Programs had to be as compact and efficient as possible. Programs were read from**[**punched cards,**](https://en.wikipedia.org/wiki/Punched_card)**which had an obvious finite width (typically, 80 columns). You couldn’t type past the end of the line on a punched card.**  Wherever space could be saved, it was. An easy---and, therefore, common---trick was to store year values as two digits. For example, someone would punch in 66 instead of 1966. Because the software treated all dates as occurring in the 20th century, it was understood that 66 meant 1966.  Eventually, hardware capabilities improved. There were faster processors, more RAM, and [computer terminals](https://en.wikipedia.org/wiki/Computer_terminal) replaced punched cards and [tapes](https://en.wikipedia.org/wiki/Punched_tape). Magnetic media, such as tapes and hard drives, were used to store data and programs. However, by this time there was a large body of existing data. | |  |  | | | | | |
|  | Picture Caption: To make your document look professionally produced, Word provides header, footer, cover page, and text box designs that complement each other. | | | | | |
| Computer technology was moving on, but the functions of the departments that used these systems remained the same. Even when software was renewed or replaced, the data format remained unchanged. Software continued to use and expect two-digit years. As more data accumulated, the problem was compounded. The body of data was huge in some cases. Making the data format into a sacred cow was another reason. All new software had to pander to the data, which was never converted to use four-digit years.  Storage and memory limitations arise in contemporary systems, too. For example, [embedded systems](https://en.wikipedia.org/wiki/Embedded_system), such as firmware in routers and firewalls, are obviously constrained by space limitations.  [Programmable logic controllers](https://en.wikipedia.org/wiki/Programmable_logic_controller) (PLCs), automated machinery, robotic production lines, and industrial control systems were all programmed to use a data representation that was as compact possible.  Trimming four digits down to two is quite a space saver---it's a quick way to cut your storage requirement in half. Plus, the more dates you have to deal with, the bigger the benefit | | | | | |
|  | | |  | | |
|  | |  |
| Picture Caption: To make your document look professionally produced, Word provides header, footer, cover page, and text box designs that complement each other. | |
|  | | | | | | | | |
|  | | | | Page 1 |  | | | |
|  | | | |  | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | |
| If you only use two digits for year values, you can’t differentiate between dates in different centuries. The software was written to treat all dates as though they were in the 20th century. This gives false results when you hit the next century. The year 2000 would be stored as 00. Therefore, the program would interpret it as 1900, 2015 would be treated as 1915, and so on.  At the stroke of midnight on Dec. 31, 1999, every computer---and every device with a microprocessor and embedded software---that stored and processed dates as two digits would face this problem. Perhaps the software would accept the wrong date and carry on, producing garbage output. Or, perhaps it would throw an error and carry on---or, completely choke and crash.  This didn’t just apply to mainframes, minicomputers, networks, and desktops. Microprocessors were running in aircraft, factories, power stations, missile control systems, and communication satellites. Practically everything that was automated, electronic, or configurable had some code in it. The scale of the issue was monumental. | |  | | | y2k bug is also called millenium bug |
|  |
|  | |  | | |  |
|  | | | | | **According to another (not so widely known) rule,**[**if a year is divisible by 400, it's a leap year**](https://slate.com/technology/2016/02/the-math-behind-leap-years.html)**. Much of the software that had been written hadn't applied the latter rule. Therefore, it wouldn't recognize the year 2000 as a leap year. As a result, how it would perform on Feb. 29, 2000, was unpredictable.**  **In President Bill Clinton’s 1999 State of the Union, he said:**  ***"We need every state and local government, every business, large and small, to work with us to make sure that [the] Y2K computer bug will be remembered as the last headache of the 20th century, not the first crisis of the 21st century."***  **The previous October, Clinton had signed the**[**Year 2000 Information and Readiness Disclosure act**](https://www.congress.gov/105/plaws/publ271/PLAW-105publ271.pdf)**.**  **This Is Going to Take Some Time**  **Long before 1999, governments and companies worldwide had been working hard to find fixes and implement work-arounds for Y2K.**  **At first, it seemed the simplest fix was to expand the date or year field to hold two more digits, add 1900 to each year value, and ta-da! You then had four-digit years. Your old data would be preserved correctly, and new data would slot in nicely.**  **Sadly, in many cases that solution wasn’t possible due to cost, perceived data risk, and the sheer size of the task. Where possible, it was the best thing to do. Your systems would be date-safe right up to 9999.** |
| Picture Caption: To make your document look professionally produced, Word provides header, footer, cover page, and text box designs that complement each other. | | | | |
| What would happen if all these systems flicked from 1999 one second to 1900 the next? | | | | |
| Typically, some quarters predicted the end of days and the fall of society. In scenes that will resonate with many in the current pandemic, some took to [stockpiling essential supplies](https://www.nytimes.com/1999/02/09/business/fear-of-the-year-2000-bug-is-a-problem-too.html). Others called the whole thing a hoax, but, undeniably, it was big news. It became | known as the "millennium," "Year 2000," and "Y2K" bug. There were other, secondary, concerns. The year 2000 was a leap year, and many computers---even leap-year savvy systems---didn't take this into account. If a year is divisible by four, it's a leap year; if it's divisible by 100, it isn't. | | | |
| Picture Caption: To make your document look professionally produced, Word provides header, footer, cover page, and text box designs that complement each other. |
|  | | | Page 2 |  | |
|  | | |  | |

You could adopt a scheme like the following:

* For a month between 1 and 12, add 1900 to the year value.
* For a month between 41 and 52, add 2000 to the year value, and then subtract 40 from the month.
* For a month between 21 and 32, add 1800 to the year value, and then subtract 20 from the month.

You had to modify the programs to encode and decode the slightly obfuscated dates, of course. The logic in the data verification routines had to be adjusted, as well, to accept crazy values (like 44 for a month). Other schemes used variations of this approach. Encoding the dates as 14-bit, binary numbers and storing the integer representations in the date fields was a similar approach at the bit-level.

Another system that repurposed the six digits used to store dates dispensed with months entirely. Instead of storing

MMDDYY

, they swapped to a

D DDDCYY

 format:

* **DDD:** The day of the year (1 to 365, or 366 for leap years).
* **C:** A flag representing the century.
* **YY:** The year.

Work-arounds abounded, too. One method was to pick a year as a pivot year. If all your existing data was newer than 1921, you could use 1920 as the pivot year. Any dates between 00 and 20 were taken to mean 2000 to 2020. Anything from 21 to 99 meant 1921 to 1999.

These were short-term fixes, of course. It bought you a couple of decades to implement a real fix or migrate to a newer system.

Revisit working systems to update old fixes that are still running? Yeah, right! Unfortunately, society doesn't do that much---just look at [all the COBOL applications](https://www.howtogeek.com/667596/what-is-cobol-and-why-do-so-many-institutions-rely-on-it/) that are still widely in use.

**Related:**[**What Is COBOL, and Why Do So Many Institutions Rely on It?**](https://www.howtogeek.com/667596/what-is-cobol-and-why-do-so-many-institutions-rely-on-it/)

**Y2K Compliant? Prove It!**

Fixing in-house systems was one thing. Fixing code, and then distributing patches to all of the customer devices out in the field was another, entirely. And what about software development tools, like software libraries? Had they jeopardized your product? Did you use development partners or suppliers for some of the code in your product? Was their code safe and Y2K compliant? Who was responsible if a customer or client had an issue?

Businesses found themselves in the middle of a paperwork storm. Companies were falling over themselves requesting legally-binding statements of compliance from software suppliers and development partners. They wanted to see your overarching Y2K Preparedness Plan, and your system-specific Y2K Code Review and Remediation reports.

They also wanted a statement verifying your code was Y2K safe, and that, in the event something bad happened on or after Jan. 1, 2000, you’d accept responsibility and they’d be absolved.

In 1999, I was working as the Development Manager of a U.K.-based software house. We made products that interfaced with business telephone systems. Our products provided the automatic call-handling professional call centers rely on daily. Our customers were major players in this field, including [BT](https://en.wikipedia.org/wiki/BT_Group), [Nortel](https://en.wikipedia.org/wiki/Nortel), and [Avaya](https://en.wikipedia.org/wiki/Avaya). They were reselling our rebadged products to untold numbers of their customers around the globe.

On the backs of these giants, our software was running in 97 different countries. Due to different time zones, the software was also going to go through midnight on New Year's Eve, 1999, [over 30 times](https://www.timeanddate.com/time/current-number-time-zones.html)!

Needless to say, these market leaders were feeling somewhat exposed. They wanted hard evidence that our code was compliant. They also wanted to know the methodology of our code reviews and test suites were sound, and that the test results were repeatable. We went through the mangle, but came through it with a clean bill of health. Of course, dealing with all of this took time and money. Even though our code was compliant, we had to withstand the financial hit of proving it.

Still, we got off lighter than most. The total global cost of preparing for Y2K was estimated to be [between $300 to $600 billion](https://www.theguardian.com/technology/2000/jan/05/y2k.guardiananalysispage) by [Gartner](https://www.gartner.com/en), and $825 billion by [Capgemini](https://www.capgemini.com/gb-en/). The U.S. alone spent over $100 billion. It's also been calculated that thousands of man-years were devoted to addressing the Y2K bug.

**The Millennium Dawns**

[Lukas Gojda/Shutterstock](https://www.shutterstock.com/image-photo/commercial-airplane-jetliner-flying-above-dramatic-1291051624)

There’s nothing like putting your money where your mouth is. On New Year’s Eve, 1999, John Koskinen, chairman of the President's Council on Year 2000 Conversion, boarded a flight that would still be in the air at midnight. Koskinen wanted to demonstrate to the public his faith in the vastly expensive, multiyear remediation it had taken to get the U.S. millennium-ready. He landed safely.

It’s easy for non-techies to look back and think the millennium bug was overblown, overhyped, and just a way for people to make money. Nothing happened, right? So, what was the fuss about?

Imagine there’s a dam in the mountains, holding back a lake. Below it is a village. A shepherd announces to the village he’s seen cracks in the dam, and it won’t last more than a year. A plan is drawn up and work begins to stabilize the dam. Finally, the construction work is finished, and the predicted failure date rolls past without incident.

Some villagers might start muttering they knew there was nothing to worry about, and look, nothing's happened. It’s as if they have a blind spot for the time where the threat was identified, addressed, and eliminated.

The Y2K equivalent of the shepherd was Peter de Jager, the man credited with bringing the issue into the public consciousness in [a 1993 article of *Computerworld*magazine](https://books.google.co.uk/books?redir_esc=y&id=PgsZJgu-zDkC&q=peter+de+jager#v=onepage&q&f=false). He continued to campaign until it was taken seriously.

As the new millennium dawned, de Jager was also en route on a flight from [Chicago to London](https://www.computerworld.com/article/2597264/have-we-learned-nothing-from-y2k-.html). And also, just like Koskinen’s, de Jager's flight arrived safely and without incident.

**What Did Happen?**

Despite the herculean efforts to prevent Y2K from affecting computer systems, there were cases that slipped through the net. The situation in which the world would have found itself without a net would've been unthinkable.

Planes didn’t fall from the sky and nuclear missiles didn’t self-launch, despite predictions from doom-mongers. Although personnel at a U.S. tracking station did get a slight frisson when they observed the launch of [three missiles from Russia](http://news.bbc.co.uk/1/hi/sci/tech/585013.stm).

This, however, was a human-ordered launch of three [SCUD missiles](https://en.wikipedia.org/wiki/Scud) as the Russian-Chechnyan dispute continued to escalate. It did raise eyebrows and heart rates, though.

Here are some other incidents that occurred:

* Two Nuclear power plants in Japan developed [faults that were quickly addressed](http://news.bbc.co.uk/1/hi/sci/tech/586620.stm). The faults were described as minor and nonthreatening.
* The age of the first baby born in the new millennium in Denmark [was registered as 100](https://www.mentalfloss.com/article/610706/problems-caused-by-y2k).
* Bus tickets in Australia [were printed with the wrong date](http://news.bbc.co.uk/1/hi/sci/tech/586620.stm) and rejected by ticket scanning hardware.
* Egypt’s national newswire service [failed, but was reinstated quickly](https://web.archive.org/web/20041207152504/http:/archives.cnn.com/2000/TECH/computing/01/01/y2k.weekend.wrap/index.html).
* U.S. spy satellites were knocked off-air for [three days due to a faulty patch to correct the Y2K bug](https://www.cnet.com/news/report-y2k-fix-disrupts-u-s-spy-satellites-for-days-not-hours/).
* A man returning a copy of The General's Daughter to a video store in New York [was presented with a bill for $91,250](http://news.bbc.co.uk/1/hi/world/americas/589836.stm) for bringing the tape back 100 years late.
* Several months into the 2000s, a health official in one region of England spotted a statistical anomaly in the number of [children born with Down’s Syndrome](https://www.theguardian.com/uk/2001/sep/14/martinwainwright). The ages of 154 mothers had been incorrectly calculated in January, skewing test results. The ages of these women put them in a high-risk group, but it wasn't detected. If the risks had been correctly identified, the mothers would have [been offered an amniocentesis test](https://en.wikipedia.org/wiki/Amniocentesis). Four children were born with Down’s Syndrome and two pregnancies were terminated.

**The Legacy: 20 Years Later**

Remember those pivot years we mentioned? They were the work-around that bought people and companies a few decades to put in a real fix for Y2K. There are some systems that are still relying on this temporary fix and are still in service. We've already seen some in-service failures.

At the beginning of this year, parking meters in New York [stopped accepting credit card payments](https://www.zdnet.com/article/the-y2k-bug-is-back-causing-headaches-for-developers-again/). This was attributed to the fact that they hit the upper bounds of their pivot year. All 14,000 parking meters had to be individually visited and updated.

In other words, the big time bomb spawned a lot of little time bombs.