

# Devil's advocate: the case for the defence of the Y2K bug

Peter Moon, Melbourne

*Most people can recite the genesis of the 'Millennium Bug' like a mantra. "Computer programmers used the last two digits instead of four to denote years, so computers will read the year 2000 as 1900." The questions on many lips remain: what possessed them to take such a short-sighted approach? Why did they inflict a trillion dollar problem on our generation? Surely someone is legally liable for it all?*

## The answer lies in the economics.

It is the same reason that cars without air bags are still legal to drive. The same reason we still have undivided highways. In every case, the ideal way to go is obvious, but the cost of the fare is prohibitive.

In 1965, Gordon Moore (who co-founded Intel Corporation three years later) proposed a rule of thumb that computer processing power would double every year. The prediction proved uncannily accurate, and has come to be known as Moore's Law. From 1965 to the present, processors have become more powerful by a factor in the order of 234. That's around 17 billion times. For context, let's turn the telescope the other way around and work out what a seventeen billionth of today's processing power would be like. That is what those early programmers had to work with.

There are roughly 350 million personal computers in the world today. Divide their number by 17 billion and you arrive at 0.02. In effect then, in 1965, all the world's computers combined added up to 2% of the processing power of a single desktop PC today. If there were 5,000 programmers across the world, each had the equivalent of 0.000004 of a 1999 computer to work with. Processing a name, or a time, or a date took just as many steps as it does today, but each step took hundreds of thousands of times longer.

If the same proportions applied to salaries, John Howard would be earning 75

cents a year as Prime Minister, but paying expenses in 1998 prices. 'Couldn't be done', you say? That is a measure of the task facing a 1965 programmer asked to write a 'simple' database containing names, addresses and dates of birth. It's one reason why cutting dates from 8 digits to 6 was essential.

## The technical logic

Leaving sheer number crunching power aside, a computer needs to store data in some kind of memory. A typical lawyer's desktop PC today might sport 32 megabytes of random access memory, or RAM. In 1970, RAM cost around \$AUD 5 million per megabyte. The 16 MB upgrade you paid \$150 for last month would have set you back \$80 million then. How would you feel about a computer programmer who tossed away a few percent of that – several million dollars of your money – just to ensure the system will not be compromised in 30 years time?

As the millennium draws to a close, PCs typically store long term data on hard drives that cost about three cents a megabyte. The paper punch cards in common use for programming and data storage until the early 1970s stored 80 bytes each, so processing a megabyte required the individual punching, storage and entry of about 13 million cards. (Programmers from those times give graphic accounts of their despair when a pile of cards holding just a couple of kilobytes of data or program code were dropped on the floor and shuffled out of order.) The giant tapes we see in black and white movies were barely more efficient. The volume of data we store on a laptop hard drive filled rooms with tapes and paper cards when Gough Whitlam was elected. No wonder it was regarded as good programming practice to shave every last byte from code and data.

Way back when the Family Law Act was in its first draft, the use of two digit

representations of years was a decade old and universally implemented – out of economic necessity. The IBM personal computer was still almost a decade from being invented. Anyone who could have predicted its rise would today make Bill Gates look like a pauper.

It is unrealistic to blame programmers of the sixth and seventh decades of the century for 'short-sighted' design decisions that threaten business viability at the end of the tenth because of an unforeseen revolution that occurred in the eighth and ninth. It's like blaming Gutenberg for porn magazines or the first Viking sailors for the loss of the Titanic.

Abbreviation is still one of the most fundamental concepts in computing. Without caching, proxy serving, vector-based graphics and data compression, computing and the Internet would be decades in arrears of where they are. When it became entrenched, date abbreviation was just as sensible as any of them.

While estimating of the cost of Y2K remediation is becoming a national pastime, few commentators have considered the savings attributable to the 2 digit year format. Associate Professor Leon Kapplan of the University of North Texas made an educated guesstimate in 1996.

Kapplan concluded that an average 3 to 6 % of data comprises date information, so that you could reasonably attribute a 1% reduction in disk space requirements by reducing date fields from 8 digits to 6. Referring to US dollars, he reasoned, "In light of these savings, the \$10 to \$50 million estimated cost organizations may incur to correct the "millennium bug" pales in terms of benefits accrued. One organization, currently storing 12 terabytes of mainframe data and about 700 GB 15 years ago, estimates that if their average storage over the 30 years from 1963 through 1992 were only 10 GB, they saved over \$100 million in 1995 dollars." ►



Peter Moon



# Limited liability law 'bad'

MARK HOLLANDS

□ Year 2000

**NEW** United States legislation limiting financial compensation from Year 2000 failures could have a dramatic effect on Australian companies, a leading IT lawyer has warned.

The possibility the Australian Government could follow the US lead after copying the Washington-inspired Good Samaritan law could not be ruled out, according to Phillip Hourigan, of Deacons Graham & James.

The Year 2000 Limitation of Liability Act, passed by Congress last week, creates proportionate liability in which IT companies can be prosecuted only for a share of any Y2K damage.

Compensation would be set at \$US250,000 (\$375,000) in many instances.

The Act would affect the rights of Australian companies to recover compensation for Y2K damage caused by US-based vendors, said Mr Hourigan, who specialises in IT and intellectual property cases.

"This legislation shows the power and influence of IT companies in the US and also indicates the political realisation of the value of the IT industry to the US economy," he said.

"But the legislation has removed the rights of the consumer in the case of IT failures.

"It is designed to stop any possibility of class actions arising from Year 2000 failures," he said.

Many analysts and legal firms had been predicting the US would be awash with Y2K compensation claims next year. Some predicted IT companies could face up to \$US1 trillion in total claims, making the law cases more expensive than the Y2K remediation work.

Mr Hourigan said he would now advise any Australian company to avoid taking part in a US-initiated class action against an American hardware or software vendor.

"It is just not worth the time and effort," Mr Hourigan said.

"The legislation has been designed to stop massive pay-outs that we have seen in the past."

He argued the \$US250,000 liability cap had stripped away consumer

rights, which was unfair for a company that could be wiped out because of a long-term computer failure.

The legislation covers consequential damages, such as loss of life or injury, which would normally attract court-arbitrated damages that could run into the millions of dollars.

"We need to watch the Australian Government to see whether it will follow the US example," he said.

He described the local Good Samaritan law, passed earlier this year to encourage the exchange of information on Y2K issues, as an unmitigated disaster.

"There were too many loopholes in the Australian law. It had had little impact in convincing company executives and their lawyers that it was safe to talk openly about their Y2K experiences," he said.

American multinational companies based in Australia would not be covered by the new Act.

Local companies could sue US subsidiaries under the provisions of the Trades Practices Act or the Goods and Services legislation.

However, local businesses using US companies with no local office or distributor would have to use US courts and would not have the breadth of remedies that exist in Australia.

Mr Hourigan said the new US legislation was also aimed at curbing opportunist lawyers who were looking to create class actions from IT failures.

"These lawyers work on contingency fees and take a third of the pay-out if they win the case. It can be a pot of gold for them," he said.

"The legislation has outlawed that practice and curbed legal charges to \$US1000 an hour."

While that was not an insignificant charge, Mr Hourigan said it was little compared with the financial benefits of a compensation pay-out.

Australia did not have the same aggressive environment of litigation as the US, but companies such as Slater & Gordon were heading towards that sort of action, he said.

The Australian 13/7 1999. Reproduced with permission.

Once cost constraints had given birth to the practice of date truncation, a series of factors ensured its longevity. Consumers demand backward compatibility – the guarantee that a software upgrade or sequel will continue to work harmoniously with earlier generations of data and programs. Few things are as guaranteed to earn a programmer or software vendor a caning as an upgrade that impairs users' access to files that worked perfectly before the 'improved version' was installed. To deliver compatibility between two versions, it makes sense not to change anything that doesn't have to be changed – which for many years included date formats.

Code recycling also played a part. Software often calls for the same functionality to be replicated many times, within a single program or across many. Like lawyers who create new documents by cutting and pasting from precedents, programmers frequently re-use portions of tried and true computer code. Just as lawyers save time and control quality by relying on precedents rather than originally drafting every line, so do programmers. As many standard routines embody truncated dates, replication of the code has cast the Y2K problem far and wide.

Business priorities have as much to answer for as any other factor. If the world had waited for universities to deliver the personal computer, this article would have been produced on a manual typewriter; it's private capital that has fuelled the revolution – under the watchful eye of financial controllers. Put simply, getting funding to solve an IT problem that will manifest itself in four, three, two or even one decade is as close to the definition of 'impossible' as you'll get.

## The Unix Analogy

An interesting parallel with the Y2K problem is the 'Year 2038 issue' with Unix computers. Unix is an industrial strength multi-user operating system that still dominates mission-critical functions for large organisations. If all the computers in the world stopped tomorrow, we'd be ruing the loss of the Unix 'boxes' more than the PCs. And in a mere 39 years, if nothing is done, all the Unix machines will stop.

Unix measures time in seconds elapsed since midnight on 1 January 1970. The designers of the operating system made sure the system could count a very large number of seconds – 2,147,483,647 to be precise. Which takes us up to 18 January 2038, a date which will provide Unix with similar indigestion to the Year 2000 problem. Massive systems that control whole enterprises will fail. Government departments will cease functioning. Banks will fail.



And yet, is anyone worried about it today? Of course not. There's plenty of time to fix it. So much time that we need not even make a mental note to give the matter some thought in 2010 or 2020. Which is precisely the atmosphere that prevailed when the Y2K problem was set in concrete around 39 years ago – with the salient difference that they had no way of knowing that computing would underpin business by the time the issue was manifest.

### Liability

Two things should be clear by now. First, the Y2K problem is not a defect or 'bug'. It is a compatibility issue that arises out of a design decision that was the right one at the time. Second, its existence has been obvious for a long time. It is about as far from a latent defect as a problem can be. With some notable exceptions, 'solving' it was only ever a matter of the consumers of technology specifying – and being prepared to pay for – Year 2000 compatible products. To the writer, that suggests that IT vendors have the benefit of a strong argument that older systems and equipment cannot reasonably be expected to have been compatible, and that in the case of recently supplied ones, the express terms of the contract should be paramount. If the purchaser wanted Y2K compliance, they can and should have specified it.

That approach to the question of liability for Y2K disruption rests on the view that the non-technical community was on notice of the issue well in advance, so that there is no disparity of knowledge which should impel the courts to facilitate remedies beyond enforcement of express terms. Y2K commentators cast about for a definitive point past which ignorance cannot seriously be claimed (in the way that juries in tobacco litigation invariably accept that the arrival of printed health warnings on cigarette packets establishes the latest time at which we all knew smoking was a health hazard). An interesting candidate is 19 September 1996 – the 'Dilbert Era'.

### The Dilbert Era

On 17, 18 & 19 September 1996, 'Dilbert' – the daily cartoon of choice for tens of millions of U.S. business persons – satirised the millennium bug in a series of

strips that appeared in major daily newspapers across the United States. Renowned for its topicality and ability to strike chords of recognition among the business community, the subject matter of the cartoons suggests that by the end of Q3 in 1996, Y2K was sufficiently well publicised that it was a suitable subject for populist humour.

That the topic was then a mainstream one was more than a lucky guess by Dilbert's creator Scott Adams. Earlier in 1996, the issue had been featured in *USA Today* (in March), *Newsweek* (in June), the *Wall Street Journal* and *Forbes* (in July). Analysis of management liability – especially that of company directors – for failure to implement Year 2000 programs can fairly treat 1996/97 as the latest time when ignorance ceased to be a defence.

### Litigation

The IT consulting industry recently took its courage in its hands in *Anderson Consulting v. J. Baker Inc.*, filed in Massachusetts on 28 August 1998. J. Baker is a U.S. clothing retailer that engaged Anderson Consulting to advise it on the selection and implementation of a new computerised merchandising system, for a fee of 'several million dollars'. The system was operational by 1991. By 1998, it was clear that it required expensive remediation to achieve Year 2000 compatibility.

By letter of demand, J. Baker claimed damages for 'negligence, misrepresentation, breach of contract, breach of the covenant of good faith and fair dealing and unfair trade practice.' Andersen Consulting responded with an application to the court for a declaration that it had complied with its contractual obligations fairly, because:

- the contract provided that 'qualified leadership personnel' of J. Baker would work closely with the consultant "at every stage"
- J. Baker 'approved' the system
- "during the period up to and including 1991, it was commonplace to use two digits to represent the date field"
- "no provision in the Agreements, workplans, design specifications or testing protocols required anything other than two-digit year fields or referred to the year 2000"

- getting the system up and running as soon as possible – as instructed –brought significantly increased profits to J. Baker
- the cost of modifying software in 1991 to achieve year 2000 compliance would have rendered the project "economically unviable".

The action was resolved on 21 December 1998 by mediation, after which J. Baker announced that it had "re-evaluated its claims and is now satisfied that Andersen Consulting had met all of its contractual obligations...." According to a press release: "Andersen Consulting has not made and will not make a payment of any kind to J. Baker in connection with ... this matter."

Settlement notwithstanding, some commentators take the view that Andersen Consulting deliberately chose to litigate a case where its position was marginal, with a view to establishing the legitimacy of its core defences: "Two-digit years were the standard until very recently. The industry had no special knowledge about the problems that it would cause in year 2000. Customers also knew, and chose not to specify – or pay for – Y2K compatibility." ■

**Peter Moon** is Special Counsel practising in information technology with Jerrard & Stuk and writes a weekly technology column for the *Australian Financial Review*.



**www.sinch.com.au**

**Amicus Attorney -  
Case Management**

**Showcase -  
Litigation Support**

**Tel: 02 - 9718 4076**