

# Preventing Y2K took lots of hard work

As we wrap up 2009, let's think back to what life was like 10 years ago in December 1999. I was pondering starting a computer consulting business here in Brenham after taking a year off from the IT world that had burnt me out due to 18 years



John Deans

of intense computer work in Houston's urban jungle.

After analyzing numerous agribusiness opportunities and watching many experienced ranchers and farmers lose money during the 1999 summer of drought and intense heat, I was about to take my vet's advice to help businesses around the county with computers and networking. Thank you Dr. Bill Eikenhorst. Great idea!

Also during 1999 I was closely monitoring the coming Y2K (Year 2000) problem that had the potential to wreak havoc on computer systems all over the world. Of course soon after 1/1/2000 came, very few problems occurred and most people thought it was a non-issue or even a hoax.

Thank goodness no really big problems erupted and this was due to years of work and billions of dollars spent fixing a

real problem before a hard deadline. Yes, there was a certain amount of hype and even some hysteria, but the Y2K problem was real and the potential dangers of it not getting fixed on time were valid.

Just so you will really understand the problem we faced ten years ago you need to see what the technical problem was with the date format. Back in the 1960s and 1970s when literally millions of lines of programming code were written in the Cobol, Fortran and later C+ computer languages, disk and memory were very expensive.

To save space and money programmers used only two digits to represent the year which took only two bytes of memory instead of the full four digits taking four bytes or twice as much space.

Bit heads back then thought surely their programs would no longer be used come the year 2000 so they coded in the date fields and formats in the mm/dd/yy instead of mm/dd/yyyy. In computers the date "March 14, 1982" looked like "031482" so when math was involved to calculate date differences they

would use those column-dependent values for the arithmetic.

The problem on Jan. 1, 2000 would be a big issue trying to do date calculations between 1231199 and 010100 since the year field would be a negative number. Then, all that programming code and even some of the previously stored data had to be repaired so the math would use 12311999 and 01012000. This simple two digit to four digit conversion process took years and loads of money to complete but it was indeed necessary.

The critical nature of this problem was brought to our attention back in 1995 when the company I worked for called Parinet was contracted for a big banking project. A huge banking chain contracted us to do a scope of work document which involved editing and testing all of their banking software to verify the date fields could handle the year 2000 change over five years from then.

Our Chicago Parinet office worked on it for a couple of weeks, put some Excel spread-

sheets together, and delivered the bank headquarters the cost estimates and timelines it would take to resolve the dozen or so software packages by converting the programs and data with the four digit date.

Turns out the bank then took this number for one of their branches and multiplied it to the rest of their sites, remote banks and other offices. It quickly scaled way out of control and not only exponentially raised the cost, but also pushed the resolution timeline way past the Dec. 31, 1999 deadline.

After they realized the magnitude of the Y2K problem the bank then hit Parinet with a modified non-disclosure document which threatened us with a massive lawsuit if we breathed a word to the public on this massive problem they had. Of course this news spread privately like wildfire throughout our company, but we kept our mouths shut to people outside our IT circle.

For the next five years, we hired and put to work dozens of programmers fixing the Cobol

programming code and reformatted gigabytes of data to convert 99 to 1999. Microsoft Project was a critical tool in helping us manage the effort to make sure we hit our milestones and have 100 percent completion before the end of the century.

Efforts like ours were going on all over the world and our industry replaced or upgraded millions of lines of programming code, replaced firmware and even hardware that had the two digit date burned into the chips. We even did forklift replacement of complete networks, mainframes and data centers.

Our industry did this also because the downside could have been disastrous if critical systems like power distribution, flight control systems, and digital medical infrastructure had not be upgraded or replaced. We also feared that if there was indeed a nationwide computer meltdown, the U.S. government would have stepped in and taken over the IT industry with oversight, regulation and new bureaucracies.

On the morning of 1/1/2000 I scanned the Internet news for any type of major Y2K failures and only found a few. Our IT professionals did a very good job in fixing the problem so it did not become a disaster.

Turns out the industry did such a good job that many thought the whole thing was just a hoax to fix software and sell hardware — this was not the case.

My IT industry fixed 99 percent of the potential problems, but just to give you a taste of what could have happened, here are some things that got missed and did indeed fail due to the Y2K bug:

Britain Bank in December 1999 had software that started rejecting credit card charges.

"Critical" monitoring system of hospital failed 01/01/2000 in Toronto, Canada.

Seven nuclear generating stations experienced minor failures in monitoring systems at in the U.S.

A control system jumped ahead 35 days at a power plant in the U.S. Midwest.

Data from a U.S. defense department satellite was unusable for a few hours in the morning of 01/01/2000.

There were hundreds of other small problems like those five above that were resolved, but do you math yourself on what would have happened if the IT industry did nothing to fix the Y2K date problem during the 1990s.

Bottom line: The Y2K problem may have been a non-event for most people and we in the IT industry was hoping for just that outcome.

Next week's column: 2009 wrapup and review.

John Deans of DeansConsulting.com is a Brenham area computer networking consultant who can be reached at 289.2233 or John@DeansConsulting.com for questions and comments.