

YEAR 2000 AND THE NRAO

Alan Bridle

An article written for the NRAO employee newsletter, *The Point Source*
October 24 1997

There are some big parties planned around the world for New Year's Eve, 1999. The top party spots were booked solid years ago. Unfortunately, a real party pooper may also show up at the stroke of midnight.

Its name is the "Millennium Bug", or "Y2K" (shorthand for Year 2000). It's not something out of science fiction, but a computer bug that could be big trouble about 800 days from now.

Take a look at your checkbook. Where you write the date on each check, it probably has a '19' pre-printed before the year. You don't have to write '1997' in full, just '97'. That saves a few seconds on every check you write. You probably thought it was a good idea.

Much of the world's computer hardware, programs, and databases have done just that for decades. The year is stored as 'YY', used as if it means '19YY'. Not keeping track of what century it is made economic sense decades ago, when data went into computers on punched cards and when "mainframes" had less disk space than today's "notebooks". It's been estimated that not storing all those 19's saved around \$30 billion, worldwide.

Now that 2000 is approaching, the real bill for this short-cut is coming due. Critical computers, programs and databases that use dates but don't know which century it is will have to be retired, refurbished, or replaced. The cost has been estimated at \$300 billion to \$600 billion worldwide.

It's not just the cost that may be painful. Because so much has to be fixed simultaneously and fixed soon, there may not be enough people to do the whole job in time. 2000 may arrive before the world's information infrastructure is fully "Y2K compliant", i.e., guaranteed to process all dates and time intervals correctly after 31 December 1999.

Why does this matter? Here's an example.

A company that recently won a 5-year Defense Department contract ending in 2002 got a 95-year delinquency notice 90 days later. A DoD computer expected deliverables in '02'. It interpreted that as '1902', compared it with '1997', then issued the 95-year notice. Nobody checked it. If that seems funny, it might be less so if it was your federal tax status being questioned by the IRS, claiming 95 years' worth of penalties!

Common sense says this shouldn't happen, of course. But computers don't have any common sense. They do exactly what we tell them, no more and no less. And what many have been told about dates makes no sense after 31 December 1999.

The problem is serious because increasing "efficiency" took people, and common sense, out of the loop in many business transactions. We rely on computers to compute bills, interest, due and delinquency

dates, etc. quickly and correctly. If they start making mistakes from century confusion, then most banks, government agencies, and businesses don't have the staff to second-guess them before problems proliferate. The computer date problems must be fixed.

You may be asking "how can using two more digits in dates be such a big problem? Just make them all YYYYMMDD instead of YYMMDD ... it's not rocket science."

Here's an analogy to show why fixing one little bug can still be big trouble.

Suppose you have copper plumbing at home, and one soldered joint starts to leak. It's no big deal, you fix it. You might call a plumber if you don't have a blowtorch and solder handy.

Now suppose that you were told that at midnight on 31 December 1999, every soldered joint in your home will come apart unless you redo them all first. You wouldn't wait until that morning to start the job! Just finding all the joints could take a while. Some might be inaccessible. You'd likely need a plumber, and to plan ahead.

But what if we found out that all the soldered joints in all the copper plumbing all over the world would come apart at midnight on 31 December 1999? That would be a crisis. Do we have enough plumbers? Enough blowtorches? Enough solder? Whose plumbing would get fixed first? The "most important" buildings (who decides which they are?), or the ones whose owners call the plumbers first? If time was running out, we might start the work in the places that make solder and blowtorches, and plan to shut off the water elsewhere.

Each Y2K bug is like one soldered joint, trivial to fix by itself. But we do have a situation where the whole world has to make the same repairs at once, with a fixed deadline. Thirty or forty years' worth of accumulated bad practice now has to be corrected in less than 800 days. This is why the Millennium Bug is dangerous: it might overwhelm the resources available to fix it in time. It is also being fixed on a first-come, first-served basis, not (yet) as part of any co-ordinated approach.

The Millennium Bug can also live in things that we don't usually think of as computers.

There are "embedded chips" in controls that regulate heating, ventilation and air conditioning in high-tech buildings; in industrial production lines; in systems that call for maintenance if they aren't serviced or calibrated on schedule (elevators, hospital equipment); in building security and safety systems; in telephone switches; in water treatment plants; in electric power generation and distribution systems. Many embedded chips use primitive date processing and may interpret '00' as '1900'. In many cases this won't matter, as the correct date is not critical to the chip's work. But in others, it will matter. These must be identified, among the seven billion or so embedded chips that are installed annually, then they must be replaced.

Unexpected Y2K failures have already been found in electric power generation (a coal-fired power plant shut down in a Y2K test because its smokestack scrubbers thought they had not been serviced for decades), in retail sales (the checkout and inventory system of a store in Detroit failed repeatedly when reading debit cards with '00' expiry years), in date-aware "uninterruptible" power supplies (some models shut off power completely when told that the year is '00'), and in date-aware pharmaceutical production lines.

The observatory, and each one of us, depends on services such as electric power, communications, banking, and retirement investments. Production and distribution of many commodities and products also takes a "just-in-time" approach that relies on accurate forecasting of needs, and on all supplies being delivered when needed. Century confusion in even a few places could therefore have serious ripple effects elsewhere.

It's clear that many vital services and supplies would be disrupted if the Year 2000 arrived tomorrow instead of two and a quarter years from now. Only a few percent of the total Y2K conversion job worldwide has yet been done.

It's unclear how close to full Y2K compliance the banks, stock exchanges, insurance companies, hospitals, governments (federal, state and municipal), air traffic control centers, railways, automated warehouses, pharmacies, electric utilities, telephone systems and industrial plants will be when Saturday, 1 January 2000 actually comes.

The Millennium Bug is serious business because these are not things that we're used to being unclear about. That's why some CEO's, and computer programmers, may be skipping the big millennium party.

What are we doing about Y2K at the NRAO?

We're now assessing where we may have Y2K problems in the hardware and software that are most critical to running the observatory. We're focusing our initial effort on the observatory's fiscal, payroll and personnel processes, and on telescope operations. We're also looking at systems that contain embedded chips, including our communications systems and building controls, as well as at computers and software.

Time is such a fundamental parameter in astronomy that the telescopes, observing software and data analysis programs must be able to go smoothly through the century transition. Most were designed to do so and we should need only to test that they do, and correct any unexpected problems that such tests reveal. We know that others will need some modification to be fully Y2K compliant.

As we do not use large amounts of date-aware computer software elsewhere in the observatory, we should not face the severe problems that confront many businesses and financial institutions. But we are not immune to them, either, and we have begun to review our own Y2K compliance and that of major suppliers of services on which we depend.

A working group has been set up to identify and help mitigate any Y2K problems within the NRAO. Each NRAO site and major division has a representative on this group: Richard Simon and Gareth Hunt (Computer Division), Tony Beasley (AOC/VLA), Carl Bignell (Personnel), Alan Bridle (Charlottesville), Jim Desmond (Fiscal), Jeff Hagen (Tucson) and Bob Vance (Green Bank). If you're concerned about a Y2K issue in any date-aware device that is essential for your work at the NRAO, please contact one of these people.

We aim to complete our initial assessment and testing of critical systems at the NRAO by the end of 1997. Only then will we know the scale of any Y2K problems within the observatory and thus be able to decide how best to handle them.

To increase awareness of Y2K issues at the NRAO, talks are planned for each site, and a Y2K Web page

has been started at <http://www.cv.nrao.edu/y2k/>. This Web page will lead you to information on the Internet about Y2K preparedness outside the NRAO, as well as to our internal Y2K documents.