Y2K Memo Series: No. 1

To: AD Distribution, M. Petty

From: Richard Simon

Subj: Year 2000 Compliance at NRAO

When clocks tick over from 1999 to 2000 in 2 years, 4 months, and change, many current computer systems and software, and other 'smart' hardware containing embedded microprocessors, may malfunction if not updated or replaced before that time. The simple convention of using 2 digits for the year instead of 4 has created a pervasive time bomb ticking away inside of much of the software and hardware we use. The effects of the so-called "Millennium Bug" may be widespread, and disastrous for organizations which are unprepared. It has become clear that substantial efforts are required to correct Year 2000 ("Y2K") problems. Widely quoted estimates place the worldwide costs to fix Y2K bugs at $600 billion, excluding the likely litigation costs in the aftermath of up to a few trillion dollars.

For the past month or so I have been carrying on informal discussions to assess the potential problems we face here at NRAO related to the Year 2000. Alan Bridle has been particularly helpful in focusing concern on this problem. Attached below this memo is a note from Alan which outlines some of the potential areas of Y2K exposure at the NRAO. I suggest that Year 2000 (Y2K) be an agenda item at 14 August's AD meeting. I will ask Alan Bridle to outline some of the potential problems we face, as well as to respond to any questions which the other AD's might have.

We at NRAO need to move quickly to begin an assessment of where we face real problems, and start solving those problems. We should not minimize the problems we might face: if we do nothing we will face severe disruption of NRAO operations on 1 January 2000. We face some urgency in addressing Y2K problems this year and next. As the broader economy begins to panic in mid-1998 and beyond, some of the fixes we might need could become expensive due to shortages, and hiring programming staff could become prohibitively expensive.

The broad areas where we face risks are the following:

Fiscal, Payroll, and Personnel: Much of the critical work in these areas is contracted out, but significant amounts of data collection and transmission rely on NRAO facilities. We must verify that contractors we rely on are themselves Y2K compliant, and we should develop contingency plans if problems occur. Many banks and other financial institutions are extremely vulnerable; it has been speculated that the recent wave of small banks merging into larger banks has been partly fueled by the small banks' realization that they will not make it. There are also significant in-house programs which will need tested and checked; for example, the personnel office uses a locally developed data base.

Telescope Operations: Most of our online systems should be Y2K compliant, but only detailed testing can find the many likely small errors. A major national observatory recently conducted a test by setting their master clock ahead. Among other interesting effects, their calculated sidereal time was negative (and ran backwards!), and some devices thought the year was 2800, not 2000. Detailed tests at NRAO will require considerable planning to insure that we can back out of them and return to normal operations. Testing complex systems like our correlators for Y2K compliance and implementing needed fixes will be challenging.

Embedded PC's: Many of our most complex electronics systems use embedded PC's and chips which are probably not Y2K compliant. Detailed testing is required to see which systems are affected, and where updates or replacement might be needed. Here at NRAO there are numerous ATs, XTs, vanilla PCs, 8086s, 286s, 386s, etc. still in use. All of these are not Y2K compliant; the question is how important is the non-compliance.

Communications: Our phone systems and PBX's, the Intranet, the Internet, and long distance telephone services are all vulnerable. We need to review the weaknesses or potential problems in the hardware we own (for example, some of our PBX's may need updated, and many of our Cisco router boxes for networking have documented problems), and develop contingency plans if important communications services are unavailable or crippled in the first part of 2000.

Utilities and Other Key Outside Requirements: Our planning will need to include the possible (probable?) disruptions in power delivery to NRAO sites. Other services to be concerned about are listed in Alan's note below.

Computing Facilities and Software: These vulnerabilities may be the most obvious and easy to deal with, but the volume of potential problems may make them challenging. The date change will affect both the operating systems and the software we run on all our machines, including UNIX workstations, PC's, and Fiscal systems. A small example of the kind of problem faced by off-line software is the FITS file format, which is not Y2K compliant. Every single FITS reader and writer routine used by our telescopes and data reduction programs will need re-written to incorporate the new FITS standard when it becomes available.

Miscellaneous Systems: If you plug it in, turn it on, or boot it up, it might be vulnerable. Alan's initial list below is a useful starting point.

Specific Action Items for NRAO's AD's:

- We should form an Observatory-wide committee to deal with our Y2K problems. I propose that Simon, Bridle, Porter, Beverage, M. Glendenning, Hunt (ex-officio), an additional AOC member, and possibly a member from Tucson, and possibly a still-to-be-named Y2K czar be on this committee.

- Each site should name a Year 2000 Coordinator to interface with the committee and locals at each site. These Coordinators should be both technically competent and sufficiently abrasive that the Y2K efforts are pushed forward.

- The approximate outline of what NRAO must do is becoming clear:

1. Each site, in cooperation with the Observatory-wide Y2K Committee, must assess its vulnerabilities and prioritize systems and operations which are vulnerable. Note that if you plug it or turn it on, it is probably vulnerable. Reviews should be held at each site this fall.

2. Across the observatory we must perform triage to select which problems must be repaired before 1 Jan 2000, which can be tolerated if they cannot be fixed until after the magic date, and which systems, programs, and devices will be abandoned rather than repaired.

3. Detailed testing of vulnerable high-priority systems should start this calendar year, so that the resources to implement repairs can be allocated next year. 1999 is too late. As a rough guide, Y2K efforts in industry are typically 20-30% of their total spending on Information Technology for the next 3 years. At NRAO, if this level of effort is required, the impact of Y2K efforts will be substantial.

4. A detailed review of critical outside vendors and contractors must be undertaken. Alan Bridle remarked that we should look at the Y2K compliance efforts of our critical suppliers as carefully as we would look at an antenna supplier. If a vendor is uncooperative or appears unlikely to make the drop dead date, we must identify alternatives.

5. Under the leadership of the Y2K Committee, contingency plans should be developed for the most critical NRAO functions to insure that basic operations can continue. Some operational changes now could allow us greatly improved flexibility if disaster strikes. For example: suppose that even though our computing systems and ADP's systems are Y2K compliant, we are unable to process payroll because communications are down or unreliable for an extended period of time. With advance planning (for example, paying everyone with the same biweekly schedule?) doing a simplified payroll by hand at each site could keep our employees paid in the event of emergency.

Cost:

Until we have completed a detailed assessment over the coming months a detailed cost is impossible to estimate. The consensus in industry is that 20-30% of computing spending over the next three years is required to address Y2K problems. This translates into a major impact here at NRAO. Depending on the difficulty of performing essential tests and the amount of remediation required, Y2K efforts here at NRAO could have a significant impact on budgeting, operations, and scheduling of key personnel.

Alan Bridle prepared the following rough outline of specific areas where NRAO might be vulnerable (not in order of priority):

Potential Year2000 (Millennium Bug, Y2K) Exposures at the NRAO:

Telescopes

Operating systems of old on-line computers (e.g., ModComps)

On-line control and monitoring software

Embedded chips with date/timing functions

- computers
- receivers and electronics with microprocessors
- power supplies, including UPS and "smart" surge protectors
- any microprocessor-controlled subassemblies with timing functions - will clocks "hang" on 2000?
- computer controlled machine tools, especially if PC chipsets used and any date-awareness

Off-line Computers (hardware and os)

- all pre-1998 PCs will likely need century bit reset
- SunOs and Win3.1 must be replaced
- AIX uncertain, rev-dependent, will we still have it?
- Solaris, Mac and Win95 os are alleged okay (Win95 fully compliant only with to-be-released upgrade)
- Linux?
- testing may be difficult in presence of licensed software and automated file operations (have to roll dates back after tests) and on networks; but alternative may be real-time failures in Jan 2000

Software

- business division software may be more critical to NRAO operations than astronomical data processing!
- no package that does date calculations should be trusted, all packages critical to operations, purchasing, management and accounting should be tested for year arithmetic, sorting and ability to write y2k dates in output files intelligibly to other software
- any time or date-aware backup procedures should be reviewed, and any time-aware auto-deletion scripts should be disabled until proven y2k-compliant
- PC software that interrogates date and time through the BIOS may still produce anomalies even when century bit is set correctly in hardware. Some commercial packages do this, and effects are BIOS-dependent. Tests are needed in actual environments that will be used post-2000 for truly mission-critical work.
- Unix systems should have no clock problems, but software running in them may still be defective, including parts of os or network management.
- Critical time/date functions that use old assembler calls should not be trusted, but tested wherever possible
- Much off-the-shelf software from commercial vendors is not yet Y2K compliant, upgrades or patches will need to be installed and paid for
- old in-house software may suffer from same problems, only it is up to us to fix it, or replace it with Y2K-compliant packages from outside, then retrain personnel to use these

Databases

- any mission-critical databases using 2-digit years that go into calculations may need to be:

  (a) backed up
  (b) overhauled (data screen problems are an obvious "nuisance" but may be less critical)

Networking

- anything depending on pre-1998 PC's, SunOS or AIX should be tested
- Internet access may be "unusual" for a while in early 2000. Effects of possible failures in telecom systems and lack of y2k readiness in many countries connecting into Internet are hard to predict. We must expect to place high reliance on our Intranet, and not assume that the general Internet will be both available and unsaturated.
- parts of our Intranet use old Cisco routers that are not likely y2k compliant and may have to be replaced
- full testing of our Intranet may be difficult as many os are unhappy about roll-forward, roll-back

date testing (not designed to accommodate parts of file systems being dated in the future, or remote past), and because of real-time license verification issues. Partial simulation in a dedicated "sacrificial subnet" may be needed.

Buildings/Physical Plant

Our biggest exposure is date-aware embedded chips with PC-type clocks, "smart" systems that think they know day of week from calendar but which may go into strange states when calendar function has 00 in year. We should look into y2k issues re:

- automated thermostats, heating/ventilation systems
- elevators with date-aware controls
- "Smart" doorlocks with date awareness (AOC?)
- telephone systems/PBX
- faxes
- "smart" copiers with date/time accounting functions in chips
- any mission-critical time/date recording systems
- safes (do we have any that are date-aware?)

Business preparation

I gather that our business division currently uses an old rev. of J.D. Edwards software for purchasing/accounting. J.D. Edwards is one of the more y2k-aware companies but it may be imperative to upgrade our operation to their latest revs. and run it for some time to ensure full y2k compliance.

I am told that programs written in dbase III (used by personnel division) can be made compliant by turning on a "century awareness" feature, but existing databases and codes may need be converted to run with this feature. Does conversion of old records have any impact for auditing purposes?

Even if the observatory becomes fully y2k compliant in time by upgrading and fully testing critical applications, the business division will need to be aware of the state of y2k preparation at any outside agencies or suppliers who are critical to our operation, including:

Funding agencies

Banks

Payroll contractors (ADP)

Power companies (we can expect 'unusual" frequency of power outages at all sites if parts of the national power grid are unstable to millennium bug, especially as 2000 is also year of solar maximum and biggest outage in US history was flare-related)

Essential materiel suppliers, esp. fuel

Telephone companies

Insurers

Very few companies or organizations are now y2k-compliant. Most should be ready to make some statement about future intentions, very few are likely to describe their plans in detail for competitive or legal reasons. But we may need to be as critical about the state of preparedness of critical suppliers (e.g., ADP) as we would be of any major telescope contractor. Or to have contingency plans ready in case they do not comply with our needs in time. We also need to know what, if any, upgrades to our computing and software will be needed to retain compatibility with outside contractors such as ADP and J.D. Edwards.

The ability to replace vulnerable systems or services closer to 2000 at reasonable cost should NOT be assumed, as everyone in the world will be trying to do this at once! The sellers' market for remediation will like start in early 1998 as many agencies are setting Jan 1, 1999 as time to test compliance. 1999 will likely see the available supply of remediation (hardware, software, and programming services) fall far behind worldwide demand. The NRAO is vulnerable to losing programmer services as salaries in commercial shops may skyrocket as the full scope of y2k problems is revealed.

Information readily available on the Web and in congressional testimony shows that many organizations that have started y2k work find that their problems are much more thorough-going than they realized. Fixing everything date-related in the short time available (we are now catching up on 30 years or more of noncompliance across a huge array of systems and services in less than 900 days) turns out to be impossible. Some old systems simply have to be junked. Some databases forgotten. Some services curtailed.

We are looking at a "come-as-you-are" problem with an immovable deadline that is the same everywhere in the world. A few unusual priority shifts may therefore be needed to handle it!

The key for late starters (and we will be one) will be triage by management, targeting the (say) 20% of all possible problems that affect (say) 80% of the mission-critical work, and finding ways to work-around or temporarily ignore the y2k failures in the rest. Without some inventory of actual exposure areas, this triage can't be done. We need that inventory A.S.A.P.

Finally, the deadline is not only immovable but bears no relation to the size of the problem. If we find that the NRAO has a big y2k problem in a critical area, we may have no alternative but to commit all available resources to remediating it. The longer we wait to start, the worse this will get, both in terms of cost, lack of time for testing, and perhaps even availability of hardware, software and personnel to do the work.