The general problem
==========

The "Year 2000' or y2k problem is the failure to represent the century part of the date correctly, or at all, in computer hardware, operating systems and software packages, in date-aware embedded micro-controllers and microprocessors, and in digital data.

The consequences of the y2k problem in the business, financial and government worlds are enormous and beginning to attract attention. The consequences of the embedded-chip y2k problem for industrial plants and infrastructure, including the power distribution system, telecommunications and fuel supplies, are also potentially very great.

It's unlikely that all problems associated with y2k in the world around the NRAO are going to get fixed in a timely fashion. Given the interconnectedness of the global economy, and its reliance upon instantaneous transactions, it's unlikely that the early days of January 2000 will see "business as usual".

But it should be the role of this committee to ensure that the NRAO, given power, essential supplies and payroll in January 2000, can indeed perform the vital functions of its emission without any major disruption by internal y2k related problems.

I believe that means that we must first inventory, and then evaluate, by testing wherever possible, our y2k exposure at the NRAO. Starting with the most critical areas, which Paul Vanden Bout identified at the AD Meeting as telescope operations, and business operations (fiscal and personnel). Then we can move on to such important but less critical areas as engineering development, astronomical data processing, etc. (where in fact we are probably already in fairly good shape).

1. Telescope operations

The areas of POSSIBLE, (I am not saying ACTUAL) exposure to y2k in Telescope operations are:

- online computers and their operating systems
- monitor and control software,
- microprocessor-controlled electronics,
- correlators,
- communications with other systems
- essential materiel supplies

We need to evaluate whether the operation of any NRAO telescope that we expect to operate in 2000 is vulnerable to y2k problems either from within or from data or commands that it will receive from elsewhere.

Because telescope control and operation are complex processes it is unlikely that we can give any telescope a credible "clean bill of health" for y2k issues without an actual operational test.

Because tests will need some care in planning, and will take time away from other activities in order to execute, we should take a first look at y2k issues for each telescope "from a distance" before planning tests.

E.g.

- is the os in the telescope control computer capable of handling dates

beyond 2000?
- o do the telescope control systems contain any date-aware embedded chips
- o do the receivers contain any date-aware embedded chips
- o do the correlators contain any date-aware embedded chips?
- o where do any embedded chips derived their date/time information from?
- o are there known adverse consequences of resetting telescope clocks and computer system dates forward and then back again to perform a test?

The simplest type of test will be to fool the telescope into thinking that the year on its master clock is 2000 without making any explicit changes to computer system dates. (In some cases, the computer system date comes from the clock, in some cases not?).

A test of this nature was conducted recently at NOAO. It's interesting to note the consequences. At one telescope, sidereal time ran backwards. At another, a hardware error, probably present for many years, prevented the second digit in the century from being reset to anything but '8'. This hardware error had gone undetected as the century had never been reset before. The telescope control system therefore thought the year was 2800, and reset after the test to 1800.

These are examples of why we have to test, and not just assert that we expect y2k compliance because we've been clever. Software that is supposed to handle a year ending in 00 may do so in a logic loop that's never been exercised before. (I presume this accounts for the 'time running backwards' result at one NOAO telescope). Or something may be broken that simply never showed up with a 19yy date.

Since we started asking questions about y2k compliance at the NRAO, one minor example showed up in the system time in the VLA on-line (Modcomp) computers. George Martin thought this would be "safe" because it is stored in a 16-bit signed integer system date derived from the VLA IAT clock. But when he checked the assembler code for this operation he found that the year was fixed algorithmically to a "cosmetic" 19YY format. This is not the date used to control the array, but would simply be the one assigned by the Modcomps to any error messages, time-stamping printouts etc. Its operational consequences are therefore minimal, but it's a home-grown example of the sort of problem that could be widespread.

Computer operating systems now in use at the NRAO (some scheduled for upgrade or replacement) contain non y2k-compliant utilities. For example, although UNIX in principle has no clock problems until 2038 and a very robust calendar facility, IBM AIX 3.2.5 (which is running Charlottesville's main server right now), has a dozen non-compliant utilities associated with account management, timed shutdown, etc. So we need to establish whether IMPORTANT computer os utilities are y2k-compliant, independent of whether system clocks support 2000.

We also need to understand how to back out of any test that advances the system date in a control computer (rather than just the apparent date of a simulated observation) to 2000. Can we backup the state of the system before the test if the system is intolerant of files created "in the future" when it is reset? Do software licenses permit a "future" test in any case?

So I suggest we first need to inventory and review the y2k issues we might expect to have at each telescope, from a priori information, checking what system designers, operators and vendors know of expected y2k compliance. Then design and perform tests of sufficient scope to check out the major modes of operation that the telescope depends on.

If we don't do tests before 2000, we'll be doing a big one when it rolls around, and perhaps under less calm circumstances than now!

2. Business operations

We have outsourced major parts (e.g. payroll to ADS and some fiscal to J.D.Edwards). These are large companies with massive y2k exposure and awareness.

ADS was recently certified by ITAA as having the core resources necessary to address y2k issues in a timely way, but this does not say anything about how, or when, they will actually be y2k compliant. J.D.Edwards is very y2k-aware, as is evident from their website.

With outsourced services our questions need to be:

- what form does their statement of compliance take, and is it credible?
- when do they expect their services to us to be compliant
- what if any changes have to be made to our data-handling and communications to be compatible with them when they become y2k compliant

There is a problem in evaluation of vendor statements in that there is no standard for "y2k compliance". Vendors may say something is "y2k compliant" if it is possible to make the product comply, but not necessarily compatibly with the compliance conditions of other products.

In-house programs and databases used for critical business and personnel purposes will have to examined and tested in-house.

A potential problem area is that 47% of all PC's purchased from major suppliers in 1997 have firmware (RTC+BIOS) that is not fully y2k-compliant and may affect some date-aware applications even when their clocks are set properly. The widely-advertised PC clock problem is not in fact very serious, as the century bit in most PC clocks can be set correctly by a DATE operation that only needs to be done once. Whether important date-aware software packages still have y2k problems because they derive date information through a non-compliant BIOS can only be tested in actual operational environments.

The major PC suppliers, and Microsoft Windows os products, should be completely y2k compatible by 1998, though they are not compliant now.

A more significant issue for the business division may be monitoring of the y2k compliance status of agencies and suppliers on whom we rely for critical services.

There are major exposures to y2k in the banking, electrical power, telecommunications and transportation areas of the economy. The Federal Government and State Governments are very far indeed from having their houses in order, and we can not expect "normal service" in every area of the economy as 2000 approaches.

It may make sense to have contingency plans for use of our emergency power generators, and larger-than-usual reserves of essential supplies, as 2000 approaches.

3. Infrastructure

==========

I suggest that after telescope operations and business explicitly, our next highest priority is observatory infrastructure such as computer networking and building physical plant.

Our internal y2k exposures (i.e. other than loss of power, telecomm services, etc from the outside) will likely be from any date-aware embedded chips or microprocessors in our environmental control systems or telephone systems, and in the management of the NRAO intranet.

Do we have any date-aware controllers in buildings, e.g. anything that behaves differently on a Saturday or Sunday than during the regular work week? If so, how do they get their dates, and if from a calendar or clock can we test its y2k compliance?
What do the manufacturers of our phone PBX systems state about their y2k compliance? Are voice mail systems y2k-compliant?

We will likely be placing enormous reliance on the NRAO intranet in 2000, because we are already doing so. We need to look carefully at all ingredients of the intranet for y2k compliance, including routers, switches, protocols, scripts and tools. Just because UNIX os's should be y2k compliant, we don't assume that everything we run in them is, or that everything we talk to them through is. Perhaps this is an arena where a dedicated sub-net behind a "firewall" could be set up for a while specifically to test our y2k compliance but using the ingredients (routers, protocols etc.) found on our main net.

The internet technology is already y2k compliant, but we cannot expect internet ACCESS to be "normal" in early 2000.

Scientific off-line computing

This is what many of us spend most of our time doing but I think it comes last in our priorities for y2k assessment for three reasons. First, no individual task in it is critical to the running of the NRAO as a whole. Second, parts of it are the responsibility of individual staff members who use "private" software packages and only they can be fully responsible for them (once the observatory has provided a functioning network for them to connect to). Third, the observatory's main off-line computing services are already y2k-compliant or are actively being made so (e.g., FITS format, AIPS)

y2k exposures may come in real time clocks, firmware, operating systems, languages and compilers, applications software (in-house and commercial), databases, interfaces and device drivers.

Priorities for Remediation

If it turns out that we have a large y2k problem anywhere, it may be very important to distinguish things that are functionally non-compliant but only have nuisance or cosmetic value. A non-compliant date appearing in an obvious output display is not as dangerous as one that is used for date arithmetic, for sorting or for scheduling of real-time events, for example.

Action
======

We should begin assessing y2k exposure in critical areas of telescope and business operations NOW.

We should extend this effort to less critical operations areas SOON.

We should decide what can be tested, and design and schedule tests.

We should defer working on y2k things that are a mere nuisance until we have fixed potential show-stoppers, if we have any.

Then we should decide whether to retire, replace, or refurbish the critical non-compliant items.

We should also prepare to develop contingency plans for our own y2k problems, and those of essential outside suppliers, in case everything is not done in time. We need to be aware of y2k compliance at all entities with which we exchange vital data, or on whom we depend for vital supplies and services.

Most organizations that have started y2k assessments discover they have a much bigger problem than they thought, with unexpected budget and personnel implications.

This can only get worse as the deadline approaches.

Any important area in which we do turn out to have a y2k problem will require very careful management, because its delivery deadline is FIXED. 1 January 2000 comes whether we are ready or not. Software projects, especially at the NRAO, are not well-known for being delivered in time and in good shape!