# The Year 2000 Farce

Peter Errington

*We should keep the two-digit year fields when solving year 2000 (Y2K) problems. Convert the existing series of year values to a string of values that the application programs can handle without error. Then convert all year values back before the outputs are produced. This will reduce the Y2K problem to primarily a job control language exercise.*

It has been estimated that $600 billion will have to be spent worldwide to fix the Y2K problem that afflicts the computer world. Most of this money will be wasted because of a poor problem definition. It does not take a computer genius to realize this—a smart kid could figure it out. Allow me to illustrate.

SK (Smart Kid) sat on her living room sofa moodily gazing at a copy of the local county newspaper, where she had just read that her county government had estimated that it would have to spend $44 million to fix the Y2K problems for all its computer systems. (Not the whole state of Maryland, just her little county.)

SK was not a "computernick." She used word processing on her computer to write papers faster, but she was totally uninterested in the computer toys that so fascinated some of her classmates. Nevertheless, she indulged herself in a silly daydream based on what she had just read in the newspaper.

In this daydream, Dwight Eisenhower was not just the pretty-good president he was in reality, but had single-handedly ended war and hunger forever. As a result, the world revised the calendar around him. The term AD no longer stood for Anno Domini, but Age of Dwight. Year AD 1 was the year Eisenhower started planning to make his move on the presidency, formerly known as 1951.

Great, thought SK. If we had actually rearranged the calendar, we would not now be worrying about spending $600 billion on the Y2K problem. It then occurred to SK that for the computational purposes of computer programs, we could pretend that the above scenario had indeed taken place. She was unsure of the details, not having computer experience, but she was sure it could be done.

I know it can be done, and I will describe how. First, however, understand that I am only discussing batch-processing computer systems. I know nothing about the problems of elevators, heart pacemakers, and other such devices. Now let us define the real problem for batch-processing computer systems.

## A Poor Problem Definition

The four-digit year values we use every day (1998, 1999, 2000 ...) would eliminate the Y2K problem were they already integrated in our computer systems. But they are not so integrated, so we must convert the two-digit year fields in our programs to four-digit fields.

## A Good Problem Definition

The two-digit year values we use every day and which are in our databases (98, 99, which lead to 00, 01...) will cause these computer programs to malfunction when we reach the year 2000.

We could convert these two-digit values to another series of continuous two-digit values that would not cause problems with computer processing in the year 2000. The following explanation has phrasing that reflects my background as an IBM mainframer, but it contains a sound general solution.

Leave the application programs alone. From any input file that contains year values, create a temporary file in which year values are converted to a continuous series that the application program can handle, then have this temporary file be the input to the application.

For example, assume a time frame starting at the beginning of 1951. This year would be 01 for the application program, 1968 would be 18, 1997 would be 47, and 2006 would be 56. All the calculations, compares, and sorts that involve a year would work fine (assuming there is no data in this system for a year earlier than 1951, in which case, my example would have to be modified).

Then, the output data sets with year fields have to be produced as temporary files in which the year values are converted back (47 becoming 97, 56 becoming 06, etc.) before the final outputs are produced.

The only work involved here would be to change job control language to produce and handle the temporary data sets, to write the truly infinitesimal programs to convert year values, and to move the interaction with the final output devices from the application programs to the final job steps in which the year values are translated back. This would be easy for people who know what they are doing. It would be much easier than pawing line by line through a Himalayan mountain range of application program coding.

As pointed out above, the calculations, compares, and sorts in the application programs would work perfectly. Therefore, my suggested method would give a high level of confidence that is missing in articles I have been reading on the Y2K problem; in these articles,

unforeseen surprises are feared from having to change date field lengths.

Some organizations, already worrying whether they can adapt before 2000, have less time than they realize. For example, a "99" in a year field may not represent a year but instead be a flag. Or information may be entered now that pertains to the next century (a loan today may have an end date in the next century, which would cause the program to reject the information with the message that a loan cannot end before it begins. This has actually happened.)

So in addition to ease of conversion and high confidence level, my suggested approach (because ease equals speed) may allow deadlines to be met that are more stringent than some organizations realize.

I have spent considerable time trying to think of valid counter arguments to the above proposal, and those I have come up with have all been weak. The most valid of the lot is that organizations might be required to furnish other organizations with files that contain four-digit year fields. But I cannot imagine an easier programming job than accepting as input a record with a two-position year field(s) and producing an output record with additional characters "19" or "20" added where appropriate.

To summarize, it seems the world is bent on squandering untold billions for no valid reason. ◆

## About the Author

**Peter Errington** has spent his entire career in data processing, starting in 1961. He retired in 1996, having worked for three private firms and two government agencies. From 1961 to 1971, he was employed at the Southern New England Telephone Co., General Electric, and Informatics. In 1971, he joined the Agency for International Development, where he specialized in payroll and personnel systems and economic and social data banks. In 1989, he moved to the Defense Logistics Agency, where, among other things, he worked extensively on Continuous Acquisition and Lifecycle Support and warehouse automation.

3809 Archer Place
Kensington, MD 20895
Voice: 301-946-7232
Fax: 703-767-6564
E-mail C/O: bette_hill@hq.dla.mil

## Year 2000 PROGRESS

### Mission-Critical Systems of Federal Departments and Agencies

| | Assessment Completed | Renovation Completed | Testing Completed | Implementation Any | Rating |
|---|---|---|---|---|---|
| **SSA** Social Security Administration | YES | 78% | 67% | YES | A- |
| **GSA** General Services Administration | YES | 35% | 26% | YES | B |
| **NSF** National Science Foundation | YES | 33% | 25% | NO | B |
| **SBA** Small Business Administration | YES | 35% | 35% | YES | B |
| **HHS** Department of Health and Human Services | YES | 28% | 10% | YES | B- |
| **EPA** Environmental Protection Agency | NO | 33% | 28% | YES | C |
| **FEMA** Federal Emergency Management Agency | NO | 35% | 35% | YES | C |
| **HUD** Department of Housing and Urban Development | YES | 9% | 2% | YES | C |
| **Interior** Department of the Interior | YES | 43% | 0% | NO | C |
| **Labor** Department of Labor | YES | 15% | 11% | YES | C |
| **State** Department of State | YES | 25% | 0% | NO | C |
| **VA** Department of Veterans Affairs | NO | 51% | 28% | YES | C |
| **DOD** Department of Defense | NO | 40% | 34% | YES | C- |
| **Commerce** Department of Commerce | NO | 15% | 6% | YES | D |
| **DOE** Department of Energy | NO | 10% | 10% | YES | D |
| **Justice** Department of Justice | YES | 1% | 1% | NO | D |
| **NRC** Nuclear Regulatory Commission | YES | 0% | 0% | NO | D |
| **OPM** Office of Personnel Management | YES | 3% | 0% | NO | D |
| **Agriculture** Department of Agriculture | NO | 8% | 4% | YES | D- |
| **NASA** National Aeronautics and Space Administration | NO | 8% | 7% | YES | D- |
| **Treasury** Department of the Treasury | NO | 6% | 5% | YES | D- |
| **AID** Agency for International Development | NO | N/A | N/A | N/A | F |
| **DOT** Department of Transportation | NO | 0% | 0% | NO | F |
| **Education** Department of Education | NO | 0% | 0% | NO | F |

Prepared for Subcommittee Chairman Steven Horn.
The departments and agencies are responsible for the accuracy and consistency of percentages reported.
Created: Sep. 15, 1997
Revised: N/A
Subcommittee on Government Management, Information, and Technology
Source: http://www.house.gov/reform/y2k/970915score.htm