



Preparing for the New Millennium

Lt. Col. Joe Jarzombek

U.S. Air Force ESIP Director



A time bomb is ticking inside billions of lines of code: those two missing digits from the calendar year designation. Most computer systems designed in the past few decades refer to the year only by its last two digits (95, 96, 97, etc.). Originally a cost-cutting measure (because computer memory was expensive and programmers looked for any way to save space), this standard practice prevents systems from distinguishing one century from another. When the rest of us make the transition from Dec. 31, 1999 to Jan. 1, 2000, many unmodified systems will turn the clock back to Jan. 1, 1900.

We are now a scant two years away from that apocalyptic date. Despite warnings, technical descriptions of the problem, solutions, and widespread awareness, the question still remains: Are we prepared for that first millisecond after midnight on Dec. 31, 1999?

At a recent Air Force year 2000 (Y2K) conference, Lt. Gen. William J. Donahue stated that he has spoken to every Air Force four-star general and every functional chief regarding the Y2K problem and has concluded that

“Some Air Force activities are in denial, others are on top of it. We will not fix everything, but we must maintain mission capability. The Air Force has \$350 million for annual software sustainment. If you do nothing but fix Y2K with your funding, so be it.”

William Ulrich and Ian Hayes, Y2K consultants, stated,

“Procrastination, whether inadvertent or by design, is preventing many organizations from launching more than one or two projects at a time. By the time companies reach full-scale deployment, it may be too late to stabilize mission-critical systems. The reasons behind this delay include analysis paralysis, politics, the fear of mistakes, confusion, lack of budget appropriation, or just ignorance of the effects of the problem. ... The arrival of the year 2000 will be death by a thousand small cuts. A critical system failure may occur from time to time, but the more common situation will involve hundreds of inconveniences that pile up day after day.” (“The Year

2000 Crisis: State of the Industry,” *Software Magazine*, October 1997).

Many factors may contribute to a Y2K system failure, including denial, procrastination, fear, confusion, and budget constraints. It is interesting to note that many of these factors are not technical in nature, but instead are psychological or programmatic.

As we prepared this issue of *CROSSTALK*, it was enlightening to see that some agencies are making real progress in becoming Y2K compliant, such as the Defense Logistics Agency (see p. 11). On the other hand, it was disheartening to hear from experts in the field that there are real risks in delivering Y2K projects on time (see “Throwing down the Y2K Gauntlet” by Peter de Jager, p. 5). The loss of a magazine’s cover art is insignificant compared to the loss of accurate missile guidance systems.

The Department of Defense is using a five-phase approach to become Y2K compliant: Awareness, Assessment, Renovation, Validation, and Implementation. If you find yourself involved in one of these phases, perhaps this issue of *CROSSTALK* will be an aid to you. You may also want to check the Web sites referenced on page 21. ♦



Better Planning and Cooperation Needed for Open Systems to Work

I enjoyed the November publisher’s note, “Open Systems Obstacles,” by Reuel Alder, and I fully agree. The way our governmental agencies are organized leads one to falsely believe that open systems are the best solution. An area that greatly hinders the open system concepts, yet is used as the prime method to meet the future and foster the open system, is what I call “the power play” with and within organizations and management (“Ours must be first if we want approval.”) Organizations don’t want to lose their identity, and manage-

ment does not want to share or lose their power base. Promotion, control, and budget often depend on this power base (“My open system is better than yours.”) This encourages waste, nonquality products, and more open systems. It limits vision, planning, development, communications, and user acceptance.

So what can we do? For one, we can start to educate ourselves on the whole rather than our piece of the pie. We should use all resources available whether they belong to us or to someone else. (Yes, the commercial world can

sometimes do it better, and two heads are better than one [usually].) We must establish measurable goals, a line of communication that encourages interfaces with vision and planning on the large, and be consistent in our vision and planning. We should stop being selfish and seek a known benefit(s) for every cost before we expend our resources, time, and budget.

Charles W. Locklin Jr.

Gunter Annex

Maxwell Air Force Base, Ala.