

## **Technology Corner: Will your system take you into the new Millennium?**

*By Jacqueline Hike*

By now I'm sure you've all been hit with the big question – What are you doing about year 2000 compliance, affectionately referred to as the Y2K issue? It's a hot topic that has received mainstream attention. Some go so far as to forecast Armageddon as a result.

### **A Little History**

Actually, the Y2K, or "date code," problem was first acknowledged as a potential concern back in the early 1980s. At that time, however, many people dismissed any idea of a possible major crisis for two reasons. One, a creative software genius somewhere in the world would develop a "Y2K silver bullet," shrink-wrap it and we could all run out to CompUSA and pick one up. Or, some expected that systems running two-digit date codes where the problem existed would be swapped out for newer problem-free software before the dawning of the new Millennium. Well, the "silver bullet" simply doesn't exist and it never will. Secondly, some systems that are as new as 1995 and still being used, fail to contain date code fixes and have been developed using the same two-digit coding convention.

### **A Shared Problem**

The general consensus is that the legacy mainframe based systems are where the majority of the Y2K problems lie. They were developed utilizing a "YYMMDD" date format designed by Cobol developers to save on what was then expensive data storage. Nevertheless, most systems with a distributed architecture, although they don't run Cobol applications, still use the same date format and present the same problems. In addition, in the client-server architectures, there is a potential of having a large number of suppliers and a distribution of assets and responsibilities. Therefore, it's a critical part of any vendor's responsibility when stating that they are year 2000 compliant to ensure that they are speaking not only of their proprietary software, but the computers, peripherals, operating systems, network, other software, and other devices as well. Moreover, even compliant systems can snag as they interoperate with older systems.

### **The Good News**

On the other hand, there may be one positive note in the Year 2000 problem. While working to take your systems into the 21<sup>st</sup> century, you could use the pause the Year 2000 furor has created to think long and hard about your existing networks.

The Year 2000 issue is one reason why PSAPs are being forced to choose between either upgrading aging mainframe systems or migrating to newer technologies, such as client-server solutions but there are other differences in these types of systems as well which may influence your decisions. Many people forecast the demise of the legacy mainframe system while others still view them as viable components of multi-platform networks. The question becomes, do you spend a lot of dollars to prepare for the future and make the necessary budget adjustments to invest in new systems based on new technologies. Or even still, do you consider your options for integrating existing mainframe systems into a distributed architecture. To decide on the best strategy, it is necessary to formulate goals reflective of your needs and future technological goals, not the limitations of your existing resources. It is also important to consider strengths and weaknesses of these types of systems.

## **The Major Differences**

Consider some of these strengths and weaknesses. To begin with, mainframe technology tends to be strongest in three areas: availability, manageability and security. But, a serious weakness is the mainframes' inability to accommodate de facto interface standards, such as Windows or the mouse-and-pointer paradigm. Probably the most significant weaknesses associated with mainframe environments are the costs associated with the development and maintenance of legacy applications. Using outdated tools for new application development increases manpower costs as well as project schedules. An often less obvious though far more expensive result of legacy development is the inherent inflexibility of the information system to adapt quickly to market changes. This inflexibility can result in the PSAP's inability to take advantage of new communications tools, for example. More importantly, it is a more critical undertaking when looking at FCC mandates and the ability to map a wireless call, for example. From an operations and logistics perspective these costs are escalated further because most developers of mainframe applications currently in use are no longer available to offer system support. These weaknesses become especially pertinent when addressing Y2K issues. Still, PSAPs have options for integrating legacy mainframe systems and creating multiple-platform environments.

What it all boils down to is that each PSAP must select the approach that best suits both objectives and short- and long-term technology strategies. In the next edition of "Technology Corner" we'll take a closer look at client-server architectures, the windows environment and the advantages the new technologies offer us as we move into the New Millennium.

*Jacqueline Hike is product line manager for Plant Equipment, Inc. Comments and suggestions should be addressed to [jhike@peinc.com](mailto:jhike@peinc.com).*