

8-15-1997

# A Logical Solution to the Year 2000 Problem?

Jane Fedorowicz  
*Bentley College*

Janis Goga  
*Bentley College*

Follow this and additional works at: <http://aisel.aisnet.org/amcis1997>

---

## Recommended Citation

Fedorowicz, Jane and Goga, Janis, "A Logical Solution to the Year 2000 Problem?" (1997). *AMCIS 1997 Proceedings*. 19.  
<http://aisel.aisnet.org/amcis1997/19>

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 1997 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact [elibrary@aisnet.org](mailto:elibrary@aisnet.org).

# **A Logical Solution to the Year 2000 Problem?**

**Jane Fedorowicz and Janis Gogan**

Bentley College

Department of Accountancy

175 Forest Street

Waltham, MA 02154

## **Introduction: Y2K Compliance Projects**

As the Millennium approaches, organizations face the Year 2000 (Y2K) problem. Countless programs, databases and chips containing internal clocks created as recently as the eighties contain two-digit year fields (e.g., "1997" = "97"). Applications coded in this way represent 2000 as "00," leading to incorrect calculations (e.g., 2000 minus 1997 should equal 3; instead, 00 minus 97 returns a result of -97). If not fixed, this software problem will give rise to a host of inaccuracies and systems failures.

Y2K compliance projects involve the following steps: hardware and software inventory; analysis (to determine which code needs to be fixed, replaced, or abandoned); conversion; testing; and migration. Note that these steps apply to all categories of software (e.g., systems software, accounting information systems, vendor packages, user-developed applications, etc.) and hardware (any computer or device - such as elevator controls -- containing processing chips). Numerous articles in the trade press have explained how organizations can conduct inventories or analyses of software portfolios, but much less attention has focused on how to perform the actual conversion.

This paper addresses a key aspect of the conversion step: the choice of approaches for converting code for Y2K compliance. Once a decision has been made to fix existing code, another decision looms: Where should changes be made? There are several alternative solutions: a) date fixes could be applied solely to the data as it resides in the file, as a field expansion (while the code logic remains the same); b) changes could be made to the programming logic only, in order to forego having to change each date in large data files, screen designs and so on; c) a combination (a) and (b). The case presented below demonstrates the complexity of this set of choices.

Given the strict project deadline posed by the onset of the Millennium, the Y2K problem comprises an unparalleled opportunity to study a broad range of issues and solution approaches within a narrow time frame, thus allowing us to cull successful efforts from situations with the same or similar technological, organizational and economic environments. This paper draws upon an extensive exploratory field study conducted in 1996 and 1997, which examined Y2K initiatives at the New York State Metropolitan Transportation Authority (MTA). Interviews revealed - consistent with prior studies of information technology initiatives - that although the choice of Y2K conversion approach was initially framed as a technical question, managers found that it was necessary to address human resource policies and strategic and organizational concerns.

## **Methodology**

The MTA was selected for study for two reasons. First, its application portfolio includes a wide variety of hardware and software technologies, from nearly obsolete to cutting edge; from large accounting information systems with extensive external interfaces to small user-developed applications to embedded chips in numerous types of machinery. This heterogeneous IT architecture was expected to elicit a broad range of interesting technical and organizational issues, and in particular, opportunities for a wide variety of code change decisions. Second, since the MTA consists of a large headquarters organization plus five semi-autonomous operating agencies, issues of within- and cross-unit organizational learning could be examined, within a single environment and time frame.

Semi-structured interviews were conducted with 5 individuals from MTA headquarters (MTAHQ) and 9 individuals from MTA's two largest operating agencies, New York City Transit Authority (NYCT) and Long Island Rail Road (LIRR). In addition, documents associated with MTA and agency Y2K initiatives were examined (including memoranda, Y2K Project Plans, Requests for Proposals, and minutes of the MTA Inter-Agency Year 2000 Work Group from its inception in February 1995 through February 1997), and two meetings of the Inter-Agency Year 2000 Work Group were observed. Interview and meeting transcripts were produced from these efforts, and were augmented by an extensive literature review.

### **Findings: Conversion Options at the MTAiiii**

A Millennium Project Update, prepared in February 1997 for the Finance Committee of the MTA Board of Directors, predicted that the cost to repair systems at MTAHQ and the five agencies would be \$25-30 million. As of February 1997, inventory and analysis of mainframe and midrange systems was complete at MTAHQ, LIRR, and NYCT, but not at the three other agencies (Metro-North, Bridges and Tunnels, and Long Island Bus). This paper focuses on those units which had completed the inventory and analysis steps.

MTAHQ and each agency had its own Y2K project team, and representatives from each project team met every 1-2 months as an All-Agency Year 2000 Work Group to address issues of common concern. One item for discussion at the first meeting of the Work Group, in February 1995: Is it necessary to expand all year fields to four digits? An MTA programmer suggested that changing the application software logic might be more efficient and less risky than expanding all date fields. He explained:

"For a physical change, you must change programs, copylibs, data dictionaries, VSAM files, job control language, reports, screens, on and on. With a logical approach, you just change your program and perhaps your sort routines -- it will be faster and cheaper."

Although this suggestion was noted in the minutes for that meeting, apparently in 1995 most participants nevertheless planned to do field expansion, which was viewed as tedious but simple and permanent. However, once each agency analyzed their systems (using automated date-analysis software), they reportedly began to appreciate the magnitude of the conversion effort, which would involve millions of lines of code and thousands of report screens. This led to the recognition that MTA would have to hire significantly more programmers (or temporary help) in order to make all these changes - or reconsider the logic approach. One Work Group member recalled: "We initially thought we would change all dates, but as we got into it we realized what a big job that is! ... There is a better way: change the software logic."

The programmer mentioned above prepared a white paper laying out arguments for and against the logic approach. Discussing it at an April 1996 Work Group meeting, he observed, "With the physical approach you must put all related programs into production at one shot, while with the logical approach you can put programs into production one at a time. If something goes wrong you can isolate it. To me, logical is much easier and safer." A manager at NYCT believed that this ability to do phased-in migration and testing was compelling. He noted, "We are a 24X7 operation; we can't afford to shut down in order to reorganize program files and libraries. I'm a big believer in the logic approach." He added, "Use logic as much as you can, because it will reduce costs. Consultants say that expansion is better, but that's because they make more money with it, and they know that a logic approach requires company- and application-specific knowledge."

Still, not every member was convinced that logic should be used so extensively. One manager cautioned:

"The logic approach is not right for everything. If dates are parts of keys in your database, then absolutely not. Also, systems that span centuries - (such as) pensions -- probably require expansion, as do systems that interface with a lot of external parties. "

With each discussion came a gradually unfolding realization that what seemed to be a purely technical choice -- software logic versus date field expansion -- had implications both for other technical choices -

such as which conversion tools to utilize - and for human resource requirements and policies, organizational structures, and management controls. Interviewees reported that these interrelated concerns had to be addressed coherently in order to keep this phase of the Year 2000 project on track.

As of February 1997, 18 of 110 mainframe applications had been converted at NYCT and LIRR, and 14 others were in the process of being converted. Within this group, approximately 80% of the program conversions entailed the logic-only approach. Plans were being formulated to convert midrange systems and user-developed applications, to replace microprocessors embedded in control devices, and to test upgraded vendor packages. The Millennium Project Update noted three concerns:

- " Staff and project management turnover has increased ... To address this, agencies are developing longevity/retention programs and planning for salary reviews as appropriate...."
- "MTA has limited control over Year 2000 conversion schedules for third party software vendors..."
- "MTA has limited control over Year 2000 conversion activities at outside companies with whom it shares information electronically, such as banks and suppliers. MTA will establish mutually acceptable schedules with these outside companies and monitor their project plans closely."

Our analysis produced a set of guidelines for selecting the appropriate conversion mechanism. Figure 1 summarizes the issues to be examined in making a decision concerning the appropriateness of each approach, given the characteristics of the program for which the change is anticipated.

### Summary and Conclusions

This paper examines the challenges addressed by a complex organization in its efforts to meet the impending Year 2000 deadline. Technical and non-technical issues surrounding the choice of software conversion techniques were addressed. This process remains ongoing at the MTA; thus, a full assessment of costs, benefits and organizational implications of these decisions cannot be included here. We will provide an update on this project during the conference presentation.

	<b>Logic</b>	<b>Expansion</b>
Date fields	Most do not change; some will (e.g., birth dates)	All date fields must change. Where packed or in binary format, referencing these fields or parts of fields must be modified.
Copylibs, data dictionaries, documentation	Most need no change.	All must change.
File sizes, DASD	No change.	Need duplicate datasets. New regions must be created.
Sort routines (i.e., DFSORT)	Do not handle Y2K logic.	No problem.
Ability to use archived data	Yes	No (problem if regulations specify data archived in original form).
Coding errors	Since fewer changes, fewer errors.	Since more changes, more errors.
Time to fix programs	Faster- easier to isolate problems.	Slower - harder to isolate problems.
Migration to production	Much less re-compiling.	Where copylib or file is changed, program must be recompiled.

	Can be phased in.	Must go "big bang."
Interfaces	Downloads may be a problem.	Downloads probably not a problem.
Application design, stability	If stable, modular, well-structured, use logic.	If undergoing debugging or revision, use expansion.
Application knowledge	High need, so use in-house staff.	Low need, so use temporary staff.
How permanent is the fix?	temporary (if dates kept for 100 or more years, logic will not work)	permanent

Figure 1: Logic vs. Expansion Approaches

**REFERENCES AVAILABLE ON REQUEST FROM THE AUTHORS.**

<sup>i</sup> For additional background on the MTA, see "Metropolitan Transportation Authority: On Track with the Year 2000 Project?", a teaching case by Janis L. Gogan and Jane Fedorowicz available through the Program on Strategic Computing and Telecommunications in the Public Sector, John F. Kennedy School of Government, Harvard University, 1997.