Preparing utility computer systems for the year 2000

Making the transition to the new millennium represents a real threat to water utility operations until a compliance program is completed.

Robert L. Ladner and Gary A. Leake

The clock is ticking toward the end of one century and the beginning of a new millennium. Throughout history, the cyclical transitions of centuries and millenniums have been viewed with varying degrees of anticipation and terror. As superstition gave way to the age of reasoning, science, and technology, people have tended to look forward to such events. Unfortunately, there is also a darker side to this rapidly approaching event: some computer hardware and software are not capable of dealing with the transition to 2000. Just one second after 11:59:59 p.m. on Dec. 31, 1999, some of the computer-based tools and control systems required daily by a utility may no longer function. Imagine the turmoil if the accounts

On Jan. 1, 2000, some 80 million computers around the world will attempt to process a 00 year. Without year 2000 (Y2K) compliance assured, some computers will crash, some programs will halt, and others will create countless errors. Worse yet, some programs will produce slightly errant or corrupt data that will be assumed valid because the errors will not be readily detected. Utilities can take strategic steps to ensure that their computer systems are immune from Y2K-related problems. This article discusses the Y2K problem and presents a methodology utilities can follow to take corrective action.

For executive summary, see page 161.

Copyright (C) 1999 American Water Works Association
receivable aging report indicates that every account balance is in arrears by more than 90 days. Suppose billing systems determined that all customers were at least 100 years “overdue” on paying their bills and started generating strongly worded letters from the billing department. What if computer systems started scheduling some, or all, customers for disconnection because of nonpayment? These are some of the year 2000 (Y2K) problems (see the sidebar on this page) that may occur with the arrival of the new millennium.

Awareness that there might be some problems began slowly in 1993 and has reached a fevered pitch. Newspapers, including The Wall Street Journal and USA Today, have published articles on the problem. Many people view it as manageable. However, the more pessimistic predict a “digital Armageddon” primarily because of a too-little, too-late response from business and government, including utilities.

Fortunately, many utilities have Y2K compliance programs that are well under way. However, those utilities that have not yet started need to act because time is running out. Utilities that wish to remain self-sufficient or lack the financial resources to acquire outside assistance can establish their own Y2K compliance programs. This article provides such utilities with easily implementable methodology.

What is the millennium problem?

The Y2K problem stems from what years ago seemed to be a benign decision to store dates inside computers using a two-digit date field, thus saving valuable memory and disk storage space. For instance, 1944, 1967, 1982, and 1999 were saved as 44, 67, 82, and 99, respectively. After Jan. 1, 2000, these two-digit years cause confusion. Does “00” represent 1900, 2000, or some other value? At the turn of the century, some 80 million computers around the world will attempt to process a 00 year. Without Y2K compliance assured, an estimated 80 percent of computers will become unreliable, and innumerable ones will crash. Some programs will halt; others will create countless errors. The situation will be exacerbated by some programs that will produce errant or corrupt data that will appear valid because the errors will not be readily detected.

A potential problem exists whenever calculations based on time or date are used within hardware or computer programs, applications, or procedures. Simple calculations are based on subtracting one date from another to arrive at the elapsed time. More complex calculations use a combination of addition, subtraction, comparison, manipulation, or database lookup. The most complex scenario includes the sharing of dates, all of which must be in the same format, between computers or systems. This type of complexity is involved every payday when paychecks are direct-deposited into bank accounts or whenever an automatic teller machine is used.

Types of Y2K-related Failures

- Training records may indicate that all staff are in arrears.
- In the area of operator certification, all operator certificates may be reported as expired.
- Security could fail, with employees no longer allowed access to facilities.
- The maintenance management information system may indicate that equipment is overdue for periodic maintenance.
- Work orders that were open during the 1999–2000 transition cannot be closed.
- Emergency response could be slowed by traffic and communications problems and lack of requisite information.
- Heating, ventilating, and air-conditioning systems could fail.
- Billing records could determine that all customer accounts are overdue by 100 years.
- Interest and penalties might be calculated for 100 years.
- Fiscal budgets may become corrupted.
- There could be no automated purchasing.
- Inspection records cannot be retrieved because of invalid date codes.
- Database searches will not return reliable information.
- Employee health care coverage might expire.
- In customer service, there might be no online assistance available for account or property information.
- Goodwill could be lost as relationships with customers become strained.
- There might be automatic time-stamping on photographs and video-inspection records.

Problems may exist wherever computer technology is used

The Y2K problem can occur in any equipment, appliances, vehicles, facilities, tracking systems, or other devices that use computer technology, including heating, ventilating, and air-conditioning systems; power and security devices; telephones; satellites; elevators; personal computers (PCs) and
mainframes; and environmental, inventory, or process control systems. It can occur in business and end-user systems, all platforms, and with electronic data interchange with suppliers and vendors. The problem could potentially adversely affect all facets of the utility, including outsourcing and design–build–operate procedures; regulatory permitting, which states may want addressed as a condition of permit renewal to avoid the consequences of water and wastewater plants dropping off-line; management responsibility; and liability. Other areas that may face difficulties include voice–data communications; network support; utility database compatibility, including historical data and reports; information sharing–linking with other organizations; geographic information systems; maintenance management and other time-dri-
ven activities; unit operations, including instruments and meters; customer service; accounts payable and receivable; and payroll.

Companies are paying big money to fix the problem

Merrill Lynch and Boeing estimate that they will spend nearly $200 million, and Washington state had committed $31 million by the end of 1997 to fix the problem. Worldwide, fixing the Y2K problem will cost an estimated $1.6 trillion. Add in the costs to respond to and resolve the lawsuits the problem is expected to generate, and costs will eventually reach $3.6 trillion, or about $621 for each person on the planet. However, a proactive program can help minimize the costs.

As of Feb. 1, 1999, 334 days remain until Jan. 1, 2000, to organize, determine the scope of, plan, implement, and test the solution.

Six myths surround the problem

Utility managers can be lulled into a false sense of security by believing any of the following myths.

It is a simple technical problem: just expand the date field by two digits. Finding all occurrences of two-digit date fields is difficult enough (e.g., the source code may be out of date or no longer available), but if many systems are involved or more than one system is sharing date data, it is a major problem to coordinate, synchronize, and reintegrate the converted systems back into production. Backed-up or archived data that contain dates need to be converted too, or a process must be created to convert the data as needed.

There is plenty of time. As of Feb. 1, 1999, 334 days remain until Jan. 1, 2000, to organize, determine the scope of, plan, implement, and test the solution. Testing will take about 50 percent of that time, leaving 167 days to do everything else. Consider the challenge of essentially testing and changing every piece of utility hardware and software within that time frame while keeping the business running.

All systems will be replaced by 2000. There is no indication that new systems are any more compliant than older systems. In June 1997, a large company undergoing a conversion experienced the failure of 47 percent of its 1997 units. Twenty-one percent of its development tools still created noncompliant code, and of more than 4,000 applications, 28 percent that were supposed to be compliant also failed.

Most companies only have new applications. This line of reasoning created the current problem. No one imagined that programs written in the 1970s would still be in use 25 years later.

An automatic solution will save the day. Automated tools and applications can help, but they are not a substitute for the hard, detailed work required to resolve the Y2K problem.

The project can be outsourced to foreign vendors. The large enterprises that started on Y2K compliance years ago have monopolized all of the foreign resources. The alternatives now available are to pay a large sum of money or have inexperienced, relatively unskilled labor work on systems.

Lack of understanding, funding have hurt progress

Most businesses are two or more years behind where they should be. In many cases, senior managers have resisted understanding the problem and making the required, incremental investment to solve it. They perceive that even after all the expense, systems will not have more capabilities than before the conversion. Another hindrance is the tendency to view the Y2K situation as an informa-
Y2K compliance requires a process

Many companies, including utilities, are taking their own steps to protect their systems. The following discussion outlines a process to move a utility into Y2K compliance and points out pitfalls to avoid.

Getting organized is critical

Y2K remedial projects must be planned and executed in an organized fashion.

Obtain management commitment. Because establishing an effective Y2K compliance program will require a substantial investment by the utility, it is imperative to have senior managers’ commitment and support. If senior managers do not understand the Y2K problem and its potential effects, education should start immediately. Discuss how failure to correct Y2K-related problems now could affect the utility’s ability to provide even limited services to customers. In addition, any problems that remain uncorrected until January 2000 may have to be fixed on-line, thus raising the difficulty, risk, and associated costs.

Establish a compliance program office. A program office should be established to lead the utility through the compliance process. The office should report to a senior manager who can ensure that program priorities are established and funding is available. The program office ensures that projects are selected and prioritized, roles and responsibilities are well defined, project plans are focused, outcomes are repeatable and predictable, knowledge and experiences are shared, productivity is improved, and all communication and initiatives are coordinated. It also functions as a clearinghouse for problems encountered by different groups, disseminates information,
develops strategy, maintains project history files, and monitors resources. The office will address the typical problems associated with a Y2K program, including contending with weak program ownership and sponsorship and working within a multiproject, limited-resource environment.

The program office can take a centralized or decentralized approach. The centralized approach enables the office to handle high volumes of system modifications and to exercise tight control. In the decentralized approach, the office still has centralized planning but establishes independent implementation teams, which enable parallelism and bring subject matter expertise to bear. Which approach is adopted should be based on the utility’s organizational and IT structure, the amount of control desired, the volume of changes, and the time available. For example, a centralized approach might be preferable if the utility is centrally managed and locates all IT functions in one place or if senior managers want to closely track the project.

Select a compliance program team and manager. Because the continued viability of the utility is at stake, the best and brightest employees who understand the utility’s business, technical, operations, maintenance, and information systems should be assigned to the team that will lead the conversion effort. Experienced, long-term employees know where to look for problems and whom to talk to about assessing them. Team members should enjoy repetitive work; pay close attention to detail (required for converting and managing code); desire contact and connectedness with all interested and affected parties; take direction well; and be team players.

Perhaps the most critical player on the Y2K team is the program manager. Because the continued viability of the utility is at stake, the best and brightest employees who understand the utility’s business, technical, operations, maintenance, and information systems should be assigned to the team that will lead the conversion effort. Experienced, long-term employees know where to look for problems and whom to talk to about assessing them. Team members should enjoy repetitive work; pay close attention to detail (required for converting and managing code); desire contact and connectedness with all interested and affected parties; take direction well; and be team players.

Compliance program should be strategic. The Y2K team will develop and guide the utility through the formal compliance program. The plan must be endorsed by senior managers and disclosed to regulatory agencies, public officials, and customers, who will want regular status updates. Because the plan may be critical to keeping the utility operating, it should have a strategic bent. A business risk assessment of legal, regulatory, and financial issues should be conducted to solidify executive sponsorship and funding, drive project prioritization, and provide the foundation for contingency plans (see the sidebar on page 41). This assessment can help the team plot out where problems are likely to occur.

The first step in developing a plan is determining what it means to be Y2K-compliant. An integral part of this step is deciding how dates will be expressed. For example, the team may decide that it is acceptable to have reports that show only the last two digits of a year, e.g., 00. If two-digit dates are not acceptable, the team will decide what conversion techniques will be used to adjust them (see the sidebar on page 44).

The team also has to determine how much time remains until compliance is required and for which systems. If the fiscal year starts earlier, it may be necessary to be ready before Y2K. These are called date-event horizons, and there may be many of them, depending on the hardware, software, and application.

The compliance program must have its own budget. This is not just another IT project that can be funded out of the IT budget. Figure 1 shows expected project costs. Creating an inventory of potentially affected systems and analyzing the effects will cost about 3 percent and 15 percent of the planning funds, respectively.

Another component of the compliance program is anticipating difficulties. For example, vendors may be unresponsive, or the pressure of trying to get everything done on time may create resentment.

<table>
<thead>
<tr>
<th>Classification</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elimination</td>
<td>The system or application is no longer needed and can be discarded.</td>
</tr>
<tr>
<td>Delay</td>
<td>Conversion of the system or application can be delayed.</td>
</tr>
<tr>
<td>Bearable</td>
<td>The consequences of not converting the system or application can be endured.</td>
</tr>
<tr>
<td>Trade-off</td>
<td>Not converting the system or application incurs less trouble.</td>
</tr>
<tr>
<td>Not mission-critical</td>
<td>Failure of the system or application does not have a substantive effect on the utility.</td>
</tr>
<tr>
<td>Mission-critical</td>
<td>Failure to convert the system or application poses dire consequences for the utility.</td>
</tr>
</tbody>
</table>
All employees must be aware of the conversion. Senior and middle managers and all staff members must be made aware of the conversion effort in easily understandable terms. Political and financial support can be garnered by educating everyone about the nature of the problem, what can be done about it, what will happen if it goes unaddressed, and how to avoid making it worse. The awareness campaign should point out that the Y2K compliance project is not only about an IT problem—it is also about managing business risk. If a key system fails and water stops flowing, this is a problem for the entire utility.

The Y2K problem should be kept visible to internal and external audiences. Brochures, newsletters, and intranets can be used to keep staff and managers apprised of activity and progress. Similar communications can maintain contact with customers, stakeholders, analysts, and the media.

Determine the scope of the project
After the project has been organized and staffed, the next step is to ascertain the size and complexity of the problem. Scope creep, or the urge to fix or improve an application while it is undergoing Y2K conversion, must be resisted. Y2K conversion is a large, complex, and deadline-sensitive process. Adding functional changes at the same time will increase costs and extend the project.

Assess the affected systems. Because addressing the Y2K problem will require the identification and review of all hardware and software, a listing needs to be compiled of all computer systems, interfaces, interaction, databases, hardware, and software that might be affected. The compliance team can use an internal survey to gather the required information, which should be stored in a repository as a corporate asset. At this point, it is necessary to at least locate and understand the meaning of all date fields, which should also be examined to see if the field is being used in a way not related to dates (see the sidebar on page 40).

Plan remedial actions
The planning can now start on how to address the problem.
Perform triage. The initial step in the planning process is to do triage on the systems with problems to separate the mission-critical systems from those less critical, determine date-event horizons, and decide whether the systems will be converted or replaced. Performing triage also identifies those systems that do not have to be corrected by Jan. 1, 2000. Systems and applications can be ranked according to urgency (Table 1).

Attention must be paid to identifying external trading partners, which includes being prepared to sever electronic communications and do business offline; identifying any non-IT infrastructure systems, such as process control systems, that may be dangerous if not converted; protecting corporate data from PC and end-user interfaces that may have not been converted; and identifying alternative suppliers for critical goods if regular suppliers are not compliant.

Locate helpful resources. A number of resources are available to help a utility involved in a Y2K conversion. Vendors of the utility's software and hardware systems should be contacted to see how compliant they are themselves and what they are doing to address the problem in their products. Resources on the Internet (see sidebar above) and software packages are available to help a water utility plan a conversion. Another resource is the experience of other utilities. Search for tools that will assist in impact assessments, code modification, and testing.

Select a conversion approach. Three conversion approaches can be used:
• Data conversion involves finding all of the date fields and changing the format of the data to whatever has been selected as the new format. Modification of the code that processes these fields will usually also be required.
• The procedural approach leaves the date fields as they are but does special pre- and postprocessing of every use of the date field in order to correctly format the date for that particular calculation.
• In a third-party upgrade, the entire system is converted to a new, Y2K-compliant system.

The procedural solution, though less costly and quicker, is riskier. Data conversion requires more time but provides a longer-term solution. These alternatives need not be mutually exclusive.

Date fields can be converted using a four-digit year, compression, encoding, integer dates, century indicator, sliding window, fixed window, or encapsulation. Procedural approaches include strict procedural and internal and external bridges. These techniques are described in the sidebar on page 44.

Points to consider when deciding between a data conversion and a procedural approach are that procedural changes can introduce more complex logic, a permanent bridge has less overhead and more accuracy than a procedural approach, and encapsulation can be confusing.

Y2K Resources on the Internet

• AWWA's Web site on Y2K problems and solutions: <http://www.awwa.org/y2k.htm>
• IBM's approach to solving the Y2K problem: <http://www.IBM.com/Year2000>
• A Web site addressing the legal implications of Y2K problems: <http://www.Y2K.com>
Techniques for Converting Date Fields to Be Compliant

- Date expansion—Replace each two-digit year field with a four-digit field (e.g., 45 would become 1945). This minimizes the potential for confusing or corrupting data.
- Century indicator—Replace each two-digit year field with a three-digit one, with the century represented by one digit (e.g., 1900–99 = 0; 2000–99 = 1). Thus, 1945, originally represented as 45, would become 045.
- Compression—Put more data in the same size field. One technique is to precede the normal character data for a date (typically six characters) with the century number (making it eight characters) and then convert to packed decimal format (storing two numbers in a single byte). Another technique stores a four-digit date within a two-byte field using hexadecimal encoding. For example, 1996 and 2000 would be represented as 07CC and 07D0, respectively. All programs that will access the date-related data will need to use the identical conversion routine.
- Date encoding—Substitute other characters for the year number to indicate the century. This is similar to the century indicator technique except the date field does not have to be expanded. For example, Sept. 12, 2001, normally represented as 010913, could be encoded as A10913. The A indicates it is the first decade of the twenty-first century.
- Fixed window—Uses a fixed year as the beginning of time for all dates. Suppose the fixed window is set to begin with 1960. Dates with a year number of 60 or larger are assumed to refer to the twentieth century. Dates less than 60 refer to the twenty-first century. Thus, 7/15/62 is really July 15, 1962, but 7/15/55 would mean July 15, 2055.
- Sliding window—This technique is similar to the fixed window, but the base year is a fixed number of years before “today.” This window typically begins 50 years ago and includes dates 50 years into the future. Thus, if today were July 15, 2004, the date 7/15/53 would mean July 15, 2053, whereas 7/15/55 would mean July 15, 1955. For this technique to work, all utility computers and systems must use the same 100-year window.
- Encapsulation—This technique is based on the fact that the calendar repeats itself every 28 years so subtracting or adding a multiple of 28 to a year number does not disturb the day of the year on which a date falls. Before processing a date, subtract 28 from the year number, do the computation, and then add 28 to the year number. This technique only works for dates between 1929 and 2027.
- Strict procedural approach—Retain the date data format and do a calculation before and perhaps after each use of the data. Before each use of a particular date field, call a subroutine to return the date in an unambiguous form; for example, the date 971107 would be transformed into 19971107.
- External bridge—Take date data in an ambiguous form (e.g., YYMMDD) and create an intermediate, temporary version that expands the date to an unambiguous form (e.g., CCYYMMDD). Then do processing on the temporary file. This technique is effective when the temporary file can be discarded, but it cannot be used in on-line and other nonbatch-oriented processing.
- Internal bridge—An internal bridge combines aspects of the procedural approach and the external bridge. Convert records from the ambiguous to the unambiguous form inside the application just before they are used (and convert back afterward, if necessary). The same subroutine can probably be used everywhere, reducing coding and testing costs, and can be easily “removed” when no longer needed.

Develop a detailed conversion plan. After the solution alternatives have been decided, the detailed plans for conversion should encompass the following steps: (1) prepare a detailed impact analysis, which includes plans to implement the selected alternatives on the various systems, (2) transition to compliant software when that alternative has been selected, (3) set standards by determining ahead of time what makes a system Y2K-compliant, (4) change all data structures with dates, (5) change test/data store to match new data structures, (6) add or remove bridges for noncompliant systems, (7) change sort parameters, (8) change code for unusual or nondate data, (9) change code to interpret dates, (10) recompile data, (11) add or remove external bridges, and (12) test the system.

Outline a testing process. All systems must be tested. Testing, which will amount to 50 percent of total conversion costs, must address three conditions: (1) whether the converted system worked correctly before Jan. 1, 2000, (2) whether it works correctly after that date, and (3) whether it works correctly during the transition from one century to the next.

Data, software, and hardware systems need to be duplicated and tested in an environment separate from production.
from the utility's production systems to avoid contaminating them. Although this may mean creating a complete duplicate system, this approach allows testing to be restarted; the data can be rolled back and the computer clocks reset to initial conditions. If a system needs to be tested but cannot be readily duplicated, special care needs to be taken to ensure that the effects of testing do not infiltrate the production systems. Today's hardware and software systems are capable of remembering the time and date even across system restarts and being cycled off and on; however, test dates that escape into the production system may be particularly difficult to expunge.

Most production systems undergo continuous change to remove bugs, improve functionality, or add new or changed data (for example, changing a customer's telephone number). Because it is unlikely that maintenance and support for production systems can be halted during the conversion, these maintenance changes must be added to systems that have already completed or are undergoing a Y2K conversion. Converted and unconverted systems must be allowed to interact for testing purposes. Additionally, to minimize parallel development problems, a system should be phased back into production after it is converted, and a strategy should be in place to address its interaction with unconverted systems.

Finally, if networking is an element of the systems being converted, an isolated network needs to be created so that spurious times from test network clocks do not leak into the production network. To date, a satisfactory way to test how well a converted system will work with other networks, including the Internet, has yet to be discovered. It appears the final test will occur in real time at the turn of the century.

Develop initialization procedures. Detailed procedures will be needed to capture the utility's existing data and load them onto the test systems. Careful planning is critical because initialization may need to be repeated if bugs in converted systems make retesting necessary.

Execute the plans

After all plans are in place, the conversion process can begin.

Initialize all systems. Test systems should be initialized with data copied from production systems. If a system fails or data are lost or corrupted, copied data will be used to restore the system to pretest condition. The copy media should be clearly labeled and segregated to avoid confusing them with routine backups and other general-use production media.

Comprehensively test the system. Most industry experts agree that testing represents the most important and time-consuming task in a Y2K program. Without thorough and proper testing, problems will go undetected until too late. Adequate resources and time must be committed to comprehensively test all software programs, modules, routines, and interfaces between computers and systems.

Document testing results. The compliance team must keep track of the test results, which should detail all Y2K problems that must be fixed, including date, time, and symptoms. The results also document progress on the conversion, provide a knowledge database, and provide an auditable record of testing and repairs.

Retest the plan. Thorough retesting is required to ensure that all Y2K problems were repaired effectively without inadvertent creation of "spin-off" problems.

Conclusion

A well-executed Y2K conversion can reap key collateral benefits. A complete inventory of computer systems will be available, and in modernizing systems and applications, utility staff members become aware of technological advances that have been made. Processes and data may be improved, and the programs in the utility's portfolio will be aligned with their purpose. IT systems will be better integrated and more responsive, and utility awareness and communications will be enhanced.

The Y2K problem represents a real threat, and each utility is at risk until a compliance program is completed. Utilities must immediately address what effect the problem will have on their particular operations. The transition to 2000 cannot be postponed.

Reference


Bibliography


About the authors: Robert L. Ladner is a senior consultant at Westin Engineering Inc., 100 Park Center Plaza, Suite 500, San Jose, CA 95113. He received BS and MS degrees and a PhD from Case Institute of Technology in Cleveland, Ohio. He is a member of the Association of Computing Machinery and the Institute of Electrical and Electronic Engineers. His previous work has been published in IT Magazine. Gary A. Leake is vice-president of consulting services for Cost Specialties Inc., 12857 Shorecrest Dr. SW, Burien, WA 98146. He is a member of AWWA.