

The Gaming Industry And The Millennium Bug

Michael L. Kasavana

Kathleen Pearl Brewer

Abstract

While much has been written about the Y2K bug in general, the gaming industry is among those slow to address the issue. As we quickly face the dawn of the new millennium, there are urgent issues to be addressed. This article examines the problem, major issues, potential solutions, and the impact of time constraints on the casino industry. **KEYWORDS:** *Y2K, Casino Technology, Gaming Technology.*

Introduction

Many Las Vegas casinos are already sold out for end-of-the-century New

Years Eve parties on December 31st 1999. While there will be much interest and excitement for millennium celebrations, the most important business question should not revolve around occupancy; it should focus on technology. Will the casino's information technology be functional the

**When the clock strikes
midnight on January 1, 2000,
will casino computing systems
know what year it is?**

morning after? Some experts warn that the "effects of the millennium bug will not be limited to the first day of the New Year. Residual fallout from corrupted data could plague reporting tools for months, maybe even years to follow." ("How prepared", 1998).

When the clock strikes midnight on January 1, 2000, will casino computing systems know what year it is? This is an important issue that casino management

*Michael L. Kasavana,
Ph.D., CHTP, School
of Hospitality
Business, Michigan
State University.*

*Kathleen Pearl
Brewer, Ph.D.,
William F. Harrah
College of Hotel
Administration,
University of Nevada,
Las Vegas.*

needs to address as soon as possible. In addition to ensuring that the casino technology is reliable, management must also be aware of casino susceptibility to the importation of damaged or tainted data as a result of contact with external computer systems (e.g., online purchasing, Internet, telecommunications, security systems, hotel property management systems).

Millennium Concerns

The main reason for concern is that programmers who developed software using older computer languages (Fortran, Cobol, and C) substituted a 6-digit abbreviation (MMDDYY) for an 8-digit date (MMDDYYYY). The original reason for abbreviated date formatting was to save precious and expensive memory space. Thus, Christmas day 1999 would be written and stored as 122599, instead of 12251999.

When it comes to correctly calculating the casino year-to-date budget, patron account balances, or inventory valuation, 2- and 4-digit years typically work equally well. For example, consider the difference between 1999 and 1976 compared to 99 minus 76. Both differences equal 23. Moving dates forward into the next century, however, changes the mathematical relationship among data since the difference between 2000 and 1976 is 24, but 00 minus 76 equals -76. The former mathematical relationship date-sensitive data simply do not transcend across the turn of the century mark. Curing the Y2K problem "requires a line-by-line rewriting of software code so computers will be able to handle four-digit references to years, and not mistake the year 2000 for the year 1900." ("Seeking solutions", 1998).

As the next millennium approaches, casino management would be wise to sharply focus on installed computer systems and the potential disaster of a year 2000 (abbreviated "Y2K") system failure. From a national perspective, it is estimated that 10% of installed business application software will require significant modification (or replacement) to ensure continuous operability. This estimate is attributable to the fact that 10% of all stored data is believed to be time/date dependent. For the casino industry, this figure is likely double, as the amount of data that is time/date dependent for internal tracking as well as regulatory compliance is extensive.

While there appears to be no universal definition of Y2K-compliant software, the "millennium bug" is expected to impact casino operations supported by non-compliant applications. The Y2K crisis, which may affect any device containing a computer chip, is of such broad scope that casino management must review the Y2K-readiness of each device, both separately and within the casino technology

The Y2K crisis, which may affect any device containing a computer chip, is of such broad scope that casino management must review the Y2K-readiness of each device, both separately and within the casino technology network.

network. Given possible connectivity to external technology, a casino faces the possibility of a Y2K ripple effect. The term 'ripple effect' describes the situation in which suppliers, business partners, patrons, and other third parties that do not cure Y2K issues can wreak havoc on casino internal applications ("The ripple effect", 1999). Systems likely to be impacted include:

Casino Accounting System
Property Management System
Restaurants and Retail Outlets
Call Accounting System
Energy Management System
Life Safety System.

The Y2K Problem

Before getting too far into the discussion, it might be a good idea to define a few 'techno' terms that have been invented or at least frequently used when describing the Y2K issue:

Year 2000-capable	--capable of being converted from non-compliant to Y2K-compliant
Year 2000-compliant	--capable of data processing and data exchange involving dates after 12/31/99
Year 2000-ready	--not Year 2000-compliant but is ready to be modified for compliance
Long Dates	--storage of 8-digit date format (MMDDYYYY)
Julian Dates	--computation of intervals based on a key date
Corrective Software	--programs to develop Y2K-compliance
Replacement Software	--substitutes for Y2K tainted software

Some interchangeable terms used for end-of-the-century data problems include:

Year-2000	--end of the century incompatibility
Y2K	--abbreviation for Year 2000 described above
Millennium Bug	--substitute for end of the century problem
Century Date Conversion	--time incompatibility

The Y2K problem is the result of two factors: 1) older computer operating systems surviving beyond a projected life cycle, and 2) the absence of century date conversion (CDC) standards in data storage design. Since operating systems tend to dictate and restrict the number of programming language options available for system development, the long-term existence of legacy operating systems can be problematic. Applying such constraints, computer programmers intentionally developed software (program code) that represented a calendar year as a two digit number (e.g., 1997 as 97), as opposed to using all four digits. At the time of implementation this approach was considered desirable and efficient. It conserved memory space and controlled overall system expenditures.

Since a 2-digit coding process only stores the last two digits of the year, the beginning of the next century presents unique computations involving the year "00".

When this two-digit number is encountered, a system failure (crash) or major data corruption appears likely. Failure as a result of improper coding may lead to the casino experiencing a significant loss of data or files. From the user's perspective, the outcomes from non-compliant systems are simple. Either the system will fail or it will continue to produce data that may not be correct. This "bad" data scenario may be the most dangerous, since decisions will be made and reports generated, including required government and regulatory reports, that may be in serious error. These errors may result in the loss of competitive advantage, productivity, profitability, and licensing.

Correcting the Y2K problem in one casino network component may compound the problem when an interfaced system device transmits transactional data that fails Year-2000 compliance standards.

Y2K Compounded

Many casinos install multiple computer systems (e.g., slot accounting system, player tracking, food and beverage, hotel property management, telecommunications) and connect them to form a casino-wide enterprise network. Correcting the Y2K problem in one casino network component (e.g., casino accounting system) may compound the problem when an interfaced system (e.g., point-of-sale system) device transmits transactional data that fails Year-2000 compliance standards. In addition, the entire Y2K casino problem can be further heightened through a lack of component synchronization. The integrity and interdependency of casino computer platforms further compounds the immensity of this potential, systemic problem. Potential Y2K problems may arise in the casino main accounting system, point-of-sale system, energy management system, timekeeping system, property management system, telecommunication system, alarm system, elevator system, and various office equipment and appliances. For example:

Patrons may receive insulting, dunning notices informing them that their casino account is 99 years past due, given erroneous accounts receivable aging calculations;

Patrons may be charged excessive amounts for overnight accommodations or other time-based events when the system calculates a usage as 99 years plus one or two hours and then applies the appropriate hourly fee; or

Patrons in an elevator may become trapped when the elevator control system miscalculates the time of its last or next scheduled maintenance.

It is important to note that all types of computers are at risk, including mid-range, desktop and laptop systems, and client/server network topologies (common

to casinos). Embedded computer chips, utilized to operate facilities equipment, such as elevator and security systems, are also prone to Y2K dysfunction.

In fact, embedded chips and systems offer additional concerns as the user is not always aware that these are separate systems or devices. The term "embedded system component" is used to describe program sections or data elements that are shared among devices or systems. They obscure themselves in hundreds of pieces of equipment and devices that are used everyday or are small systems that appear to be part of a larger system, but in fact operate independently. Any of these devices or systems that transfer date/time-sensitive data to a larger system can cause disruption, corrupt data, or crash the entire system. The Y2K problem can be compounded when corrective activities fail to resolve the problem due to a lack of detection. Any component or system that interfaces to a central system must also be 2000-compliant, or network-wide communication problems are likely to arise. In our enthusiasm to make software seamless and user friendly, we have integrated and embedded devices and systems throughout the enterprise system.

Although the cause of, and the fixes for, the Y2K problem are complicated (involving software, re-engineering and testing), the avoidance of operational problems may be even more complicated. Casino management must be keenly aware of the Y2K dilemma and play an integral role in Y2K remediation. Most computer-based clocks in casino applications are hidden and hard coded within task specific software. Consider the fact that devices in a casino network continuously commu-

nicate and maintain baseline and transactional data as time-based events. In the past, there was little thought given to differentiating between June 26, 1991 and June 26, 2001. In addition, any mathematical computations necessary to add or subtract be-

tween yearly dates are not likely to work during the rollover from 1999 to 2000. Even a computer program written to test for a specific date being later than another is unlikely to calculate correctly with only a single century of date representation.

Some programming techniques that system suppliers claim will avoid the Y2K problems are: Julian date formatting and long date formatting. Casino managers are advised to immediately contact suppliers to determine if either dating format was incorporated into installed system components.

Date Formatting

A casino supplier may claim a product is Y2K-compliant since its software incorporates Julian date formatting. A Julian dating scheme can resolve short-term Y2K problems, but the Julian approach should not be considered a permanent solution. Julian formatting relies on a five-digit coded date that can be segmented into two distinguishable data sets. The first data set involves measurement from a key

Some programming techniques that system suppliers claim will avoid the Y2K problems are: Julian date formatting and long date formatting.

date; the second set is simply the sequential number assigned to each consecutive day of a calendar year.

For example, assume the Holly Casino opened on July 6, 1957. Using the year of initial operation as the key year date and assuming that today is December 11, 1999, the five-digit Julian date of 42344 would represent this day, month, and year.

In this example, the first data set contains the number "42". This two-digit number is the mathematical difference between the key year date (1957) and the current year (1999). The second data set contains the number "344". This three-digit number represents the current day by its consecutive day-of-the-year identifier (example assumes December 11 is the 344th day of 1999).

By storing dates in an encoded Julian format, the Y2K problem is deferred until the end of the 99th year following the key year. Julian date coding consumes one less digit (YYDDD) than a standard abbreviated date format of MMDDYY and therefore provides a reasonable alternative for data storage when memory capacity is at a premium.

Similar to Julian dates, long date formatting, also called extended date formatting, is also assumed to be immune to the Y2K problem. In long date formatting, the entire month, day, and year digits are used. A long date format is 8-digits long (MMDDYYYY) and therefore does not involve any abbreviated coding or represent storage capacity savings. The long date format for December 11, 1998 would be coded as 12111998. It is important to note that this is the formatting that Microsoft has incorporated into its product line. Based on long date formatting, Microsoft has stipulated that its software and operating system products are Year-2000 compliant.

Its 8-digit length, not the order of its data contents, defines long date formatting. It is not uncommon for various programmers to employ an alternate order of contents in a long date code (e.g., YYYYMMDD or DDMMYYYY).

A second form of long date formatting involves adding an identifier digit (sometimes referred to as a smart digit) to the existing 6-digit (MMDDYY) dates stored in memory. For example,

to represent the next century, an additional digit could be attached to the stored data. Assume an identifier digit (e.g., "2") will be added as a prefix to differentiate new century data from previously stored time/date data. In this case, the format would change to a seven-digit date code of this form 2MMDDYY. The presence of the identifier digit enables the system to easily distinguish between last century and current century data.

The solution for Y2K-affected application software is to repair, re-code, replace, or retire non-compliant programs.

Corrections and Solutions

On the surface, fixing the Y2K problem appears to be a simple procedure; just isolate non-compliant software and update it. Unfortunately, locating internal shortcomings, especially where time-based computations appear can be quite complex and troublesome. There are software programs available that claim to automatically correct Cobol programs to conform to Y2K standards. If a casino system contains Cobol programming, this approach may be sufficient to eradicate the millennium bug. This procedure, however, may not apply to other legacy programming languages that historically relied on similarly abbreviated date formatting or embedded systems and processors.

A plan to attain Y2K compliance must include the following five steps:

1. Understanding of Y2K Compliance--data storage and data recognition created as 4 digit (not 8-digit) elements.
2. Developing an Action Plan--formulate a work group for determining potential issues.
3. Conducting an Inventory of Installed Systems--identify time/data dependent hardware and software components.
4. Implementing Upgrade Solutions--collaboration of vendors to discover best route for compliance.
5. Developing Contingency Plans--project potential problems and operational solutions (Elliott, 1998).

A majority of casino managers appear to be unaware or unconcerned about the potential hazards of the Y2K phenomenon. Despite the national attention the Y2K problem has received, casinos may feel immune either because they believe the situation is specific to only certain types of businesses or they have been reassured by a system supplier that the installed casino system is Y2K-compatible. In either case, there should be cause for concern, since casino operations tend to monitor time-based events (reservations, receivables, payables, budgets, etc.) through a complex network of ancillary system interfaces (point-of-sale, property management, pro shop, golf course, etc.). It is important to note that unless each component of the casino system environment is Year 2000-compliant, the overall system remains vulnerable to Y2K destruction.

The solution for Y2K-affected application software is to repair, re-code, replace, or retire non-compliant programs. Repairing systems can take several forms. A casino may choose to have all the code for legacy programs examined for date/time sensitive code and rewrite all the affected code. This is a solution that has kept many programmers busy for the past few years. Correcting the initial program code is an expensive but effective solution. A few makeshift solutions exist for patching up code. These solutions, called date-shifting and data windowing, use techniques that trick the software into thinking that one date period equals a new date period. Simply repairing or replacing selected Y2K-prone software will not likely solve the entire casino problem, since interconnected applications may not be able to interpret imported or exported data. The most effective solution is to have each software supplier certify that its product is 2000-compliant. In order to verify compliance, management should conduct a test of installed systems to verify Y2K

compliance network-wide. Although compliance stipulations on stand-alone applications can be secured more easily, the presence of interfaceable compliance is difficult to access.

The complete casino network must be checked for compliance. Casinos need to:

- determine the scope and magnitude of the Y2K problem by identifying mission critical systems;
- understand legal rights under supplier contracts;
- devise solutions for establishing compliance; and
- test interfaced systems to ensure casino-wide compliance.

An additional layer of complexity is added to the casino industry because a casino must report accurate information to a regulatory agency. In the State of Nevada for example, casinos have been asked by the Nevada State Gaming Control Board to provide a Y2K compliance report that describes applicable systems and what the casino has done to ensure that the systems are compliant.

A casino computer-based system is classified as Y2K-compliant if correct management and manipulation of time dependent data can be *demonstrated*. A casino-wide effort must be made to ensure that each individual application is Y2K-compliant and that interconnected, data exchanging applications are also Y2K-compliant. Non-compliant applications and network components must be isolated and evaluated relative to discontinuance, correction, or temporary replacement by manual processing.

Part of the procedure for proving Y2K-compliance is to request compliance verification from the system supplier. The system provider should stipulate in writing that the installed application conforms to Y2K compliance standards and provide a mechanism for the casino to test the result. Meeting this requirement can be simplified if the casino requires its system vendors to stipulate in writing that hardware and software components are Y2K compliant now ("Seeking solutions", 1998). This process is usually referred to as a Y2K-Compliance Declaration. If an application is found to be non-compliant, the supplier should be queried relative to upgrading or replacing the application to achieve Y2K-compliance. A close reading of the maintenance or support agreements may provide a basis for vendor motivation. When a non-compliant application lacks vendor support, an alternate source of service may be necessary. For proprietary systems, the casino must work to recode the parts of the systems that use, store, or accumulate date sensitive data. Since this process is time consuming, it may be past the time where this solution is reasonable.

In a best case scenario, casino management would have been wise to establish enterprise-wide compliance at least 12 months prior to the millennium. At this point careful contingency planning for both isolated and massive issues should be put into place.

Liability Issues

The Y2K problem is of such dramatic proportions that an entire legal specialty has evolved to litigate Y2K situations. Under the rubric of Technology or Intellectual Property laws, specialists have established guidelines for fiduciary stan-

dards of care and damage-recovery based upon the implications of an installed system failure due to Y2K-non-compliance.

Casino legal issues come down to three broad areas of concern: breach of contract, business interruption, and personal injury. Casino managers are advised to evaluate license agreements and service contracts to establish Y2K responsibilities. Do vendors have an obligation to provide a Y2K adjustment, modification, or replacement based upon existing contracts? If a Y2K correction or conversion is necessary, who will fix it and who will be responsible for the cost of compliance attainment? Did the software perform per its warranty? Is a vendor obligated to fix the problem? Does the casino have business interruption insurance? Is insurance sufficient enough to cover data corruption and Y2K-compatibility? Management would be wise to survey technology suppliers to determine whether installed products conform to Y2K compliance standards. It is important to note that legal experts believe that claims of nonperformance will be based on the original license agreement that accompanied the application. Management is also advised to review such documentation to determine if specific disclaimers of warranty or limited liability are relevant to a Y2K failure.

Internet Connectivity

The Internet Year 2000 Working Group conducted a study of Internet protocols to ensure Y2K-compliant data transmission. Overall, the researchers concluded that the investigation discovered little reason for concern with regards to the functionality of the protocols. With respect to other computers on the Internet, each user remains reliant upon network servers and routers to ensure compatibility ("Inside the Internet", 1999).

Summary

Shortcuts in programming over the past several decades have led to the construction of information systems with inherent programming problems. Programs written when disk space was expensive, frequently stored dates in a format that minimized the number of characters needed to capture a transactional date. Date calculations are fundamental to virtually every casino computing process. Member accounts may become bogus, vendor payments may go unrecorded, and payroll checks may bounce.

Casino managers must be aware of the potential a Y2K non-compliant application can cause the business. Management should be capable of identifying, controlling, and remedying the problem (see Figure 1 for web sites to help). Be sure the casino is prepared to raise the roof on December 31, 1999 and have it land back in place on January 1, 2000! Management must be careful not to let the millennium bug grab the casino by its database!

www.2k-times.com
www.y2k.gov

www.y2kbase.com
www.sba.gov/y2k

Figure 1. Millennium Bug Informational Web Sites

References

- Elliott, R. (1998). *Technology compliance guide for the year 2000*. Washington, DC: American Hotel & Motel Association.
- How prepared is your property for the year 2000 (Y2K)? (1998, September). *NVS International*, 11.
- Inside the Internet. (1999, January). *ZD Journals*, 6 (1), 1-2.
- Seeking solutions to the year 2000 computer dilemma. (1998, November 16). *Nation's Restaurant News, Operator Connection: Technology in Food Service*, 3 (no. 2), 5-10...25. (Insert in *Nation's Restaurant News*, 32 (no. 46).
- The ripple effect. (1999, Winter/Spring). *CIO Magazine, Y2K Special Issue*, 24.