

How india solves the y2k problem information technology essay

[Technology](#), [Information Technology](#)



Soumya NInDepartment of Computer Sciencellinois Institute of
Technologysfnu@hawk. iit. edu

Abstract-- Much focus has been paid to the so-called " Y2K" problem as we approach the end of the millenium. The programs wrote by the programmers were designed to fail in the year 2000. The current fixes could lead to the crisis in the year 10, 000, when the programs may fail again. This paper provides the brief description of the Y10K problem and solutions to the " Y10K" problem which has also been called the " YAK" problem (hex) and the " YXK" problem (Roman numerals).

Key Words—Y2K, Millenium bug, Y10K, YAK, YXK. 1. IntroductionThe predictions were made that there would be a worldwide choas on January 1, 2000, as the computers programmed with two-digit year fields would mistake it to be 1900. Many computer programs, systems and electronically controlled equipment are unable to perform correct calculations involving dates which fall after 31 December 1999. This is mainly due to the practice of using two digits to indicate the year e. g. 99 for 1999. Unless reprogrammed, bypassed or replaced these systems will malfunction at the turn of the century, if not before, with wide reaching consequences for industry, government, financial markets and many aspects of our everyday lives. The first warning of the problem was sounded by Bob Bemer, a programming expert working for IBM in 1979. It has since been called the " millennium bug" or Year 2000 computer problem and many people refer to the Y2K issue.. However, we need not wait for the turn of the century for the trouble. Signs of early troubles are already everywhere to sufficiently

warrant both IT and business managers to take the issue seriously, if they have not already done so. What seemed as a reasonable solution to costly storage problem in past, would now cost, by some estimates, 600 Billion dollars to organizations worldwide. No matter what the final cost comes out to be, even at conservative 300 billion dollars, it is no pocket change. All concern parties must have thorough understanding of the problem, its solutions, and possible ramifications. Importance of the Issue IT managers are not the only one who needs to understand the depth of Y2K problem. If timely solutions are not achieved, many business stands will lose many billions of dollars in form of lost revenues.

2. Y2K : A Millennium Bug

2. 1. What is Y2K?

The story of the Y2K bug became known to almost every inhabitant of the developed world during 1998 and 1999. During the early days of computing, programmers sought to economize on then-scarce computer storage space by writing dates with two digits for the year instead of four. Since the early days of electronic computing, almost universally, only 2 digits have been used in Computer systems to denote the year in date fields. For example, 1998 is denoted as 98. This practice was adopted to save expensive computer memory storage space and programming time. In the 60s and 70s, adding two century digits to a date field would have required storage space probably five times more expensive than that required for two - a cost difficult to justify when the general opinion was that most systems would be obsolete before the end of the century. As a result, in many applications the

Year 2000 could be interpreted as 1900 because the computer is unable to distinguish between these years which would be both denoted as 00. 2. 2. Cost implications of Y2K. The Y2K problem will possibly have many serious effects upon many of the workings of everyday life around the world. The Y2K problem, commonly known as the Y2K " bug", occurs because the computer stores the date using two digits. The computer assumes the first two digits are 19 and stores the last two. (i. e. the year 1998 would be stored as 98). What this will mean is that a lot of computer programs on which our economy depends will fail to work properly, or even to work at all. Banks may become unable to process checks or transfer funds. Governments may become unable to issue benefits or refunds. Power grids may fail, and interrupt all the services that depend on them, including water supplies. All kinds of devices with embedded processors may fail, such as elevators, printers, and navigation satellites. This can be detrimental to businesses, because if not fixed, it could deny them to access the internal system and/or produce errors in sales. Whether the company has no sales or falsely reported sales, loss is inevitable. Individuals could experience denied credit, lost funds, wrongly allocated funds, thus changing the way they do business. The most up to date, unbiased assessments predict as much as \$200 to \$300 billion dollars will be spent globally to solve the problem over a five year period. This does not include any litigation costs. So it really is serious. To place these huge numbers in perspective, \$300 billion is less than 0. 2 percent of worldwide GDP over the same period. Most economists see a small overall economic effect of Y2k - other global economic crises are far more worrisome! The Y2K problem is going to be costly for all concerned in

terms of repair costs (updating software to operate properly), in lost time, and in customer dissatisfaction. Estimates of the actual costs of Y2K vary from unknown to minimal to \$30 billion, depending on the source. Sources of these cost estimates are from private companies, many of which specialize in the Y2K problem. Searching for and changing existing code to handle a four digit date requires much manual intervention and possibly line-by-line inspection and modification. Considering the millions of lines of code which exist in most aspects of life in the United States, this compounds into a very large issue.

2. 3. Strategies used to solve the problem and their efficiencies
The primary steps and responsibly in dealing with the Y2k problem are: AWARENESS: Acknowledge there's a problem. SURVEY: Check our organizations and outside connections to identify critical systems.

Strongly encourage our ministry partners to do the same.

EVALUATE: Develop and implement a plan for evaluating which systems and connections actually have a problem. FIX: Develop and implement a plan to resolve critical issues and cope with non-critical issues. TEST: the results, well before 12/31/99! This is important not only for our own confidence in moving forward, but also for ensuring that our workers and constituents can have confidence that the organization will thrive after 1999
PLAN FOR

CONTINGENCIES: For any operational areas that give us low confidence of successful operation in 2000, develop an alternative plan that will keep us in business. This is especially important in areas where a few hours of downtime could cause a critical problem. (I. e., medical services, broadcast systems, etc.)
Several very different approaches were used to solve the Year 2000 problem in legacy systems. Four of them follow: 1. Date Expansion2-

digit years were expanded to include the century, becoming four-digit years in programs, files, and databases. This was considered the optimal solution which resulted in unambiguous dates that are permanent and easy to maintain. But, this method turned out to be costly, requiring massive testing and conversion efforts, and usually affecting entire systems.

2. Date Field Analysis

In the creation of the pattern list special care and consideration must be followed. If too many editions of date patterns are used, it would result in too many false positives. If a pattern is inadvertently left out, the possibility exists for false negatives (missing a date field). The ultimate goal is to minimize false positives and eliminate false negatives. By testing sample code from the application, the pattern list can be fine-tuned to an acceptable tolerance of false positives or false negatives. The date fields that are missed are difficult to identify during the initial search process. An essential step that aids in locating any missed dates is an analysis that occurs after completing the initial pattern search, a process known as "propagation" or "synonym" analysis. This technique discovers fields that do not have typical date field names, but are involved in date field actions (for example, MOVE WS-DATE TO WS-HOLD1). Chances are that the pattern search process would report on the entire line of code as a Y2K-impacted statement due entirely to the discovery of WS-DATE, but would not be capable of reporting on the line of code if WS-HOLD1 were then moved to another obscurely named field such as WS-HOLD2 (MOVE WS-HOLD1 TO WS-HOLD2, for example). Here we have two named date fields that would most likely be missed by the initial pattern-list search process. The potential for this type of situation requires that further analysis be done to discover any propagated or synonym fields. "

Indirects" analysis independently searches for any parent/redefines/FD relationships in regard to the previously discovered date field, and adds any discovered date fields to the list of impacted items. 3. Windowing : This involves interpreting the century date field by inserting fixed or sliding windows into the application program logic. This enables programs to process two-digit dates without converting the files. Windowing based on a "pivot date" year (Year= 30, for instance) is a reasonable approach that will: Y2K conformation should be achieved in less time. Windowing require less effort. Provide an aimed solution that will enable applications to function while plans are made to migrate the process to an environment that enables storing century values. The windowing technique is fairly generic and can be adapted to any application by adjusting the pivot year date. In Listing three, a commitment must be made to retire the application's environment by the year 2029 or else rewindow the application as the year 2030 approaches. The rewindow effort may be more straightforward if the windowing code technique stores the pivot year literal as a field name in Working Storage and common routines are used to execute the window logic. Suppose the value in POLICY-START-DATE is 290101 (January 1, 1929). In this case Listing three will conclude that the year is 2029. Now, suppose the value in POLICY-START-DATE is 300101 (January 1, 1930). Windowing analysis is only concerned with cases such as Date fields should be tested for greater than/less than conditions between fields and/or literals. Date fields should be tested for equal conditions against date literals. A value from a year should be subtracted. Actions involving reference modification. Moving century literals. Existing window logic. The primary advantage of Windowing over

expansion is that we don't have to bother about legacy data integrity other than finding the correct pivot date. But we still have to find every date-related manipulation function. For windowing to simplify matters when implementing a more permanent solution, care must be taken now in how the analysis results are documented. If done correctly, the window strategy will permit future rewindowing efforts to move the pivot year efficiently and in a timely manner. A rewindow effort would only involve revising the pivot year value stored in Working Storage, plus any unique workarounds documented during the original window implementation.

4. Date re-partitioning : In legacy databases whose size were not subject to change, six-digit year/month/day codes were converted to three-digit years (with 1999 represented as 099 and 2001 represented as 101, etc.) and three-digit days. Only I/O instructions for the date fields had to be changed, however most other date operations, and whole record operations does not require any change. This delays the eventual roll-over problem to the end of the year 2899. An effort to rewrite the system should include a complete reengineering strategy to include all of the major system development life cycle (SDLC) phases of requirement analysis, functional design, detail specifications, and so on. During this reengineering effort, existing logic may be determined useless and formulas may prove out-of-date, in addition to reports/screens that may require total removal or elimination. The Year 2K effort will give a precise inventory of all system components that will be useful as a starting point for reengineering activity. The year 2000 may very well be the largest task an organization ever faces. Selecting the right techniques and tools can significantly affect our time savings. With the

windowing method, we can achieve compliance in less time with a manageable level of effort. 3. How India solves the Y2K problem India is a late arrival to the digital world but it is not slow in solving the Y2K problem. Computers are now common in India in government, business, industry as well as homes. In view of this the experts advice the buyers to buy software's and hardwares which would run these machines with care. The buyers are expected to make sure that the computers and software's they acquire follow the Y2K requirements. This would help them avoid difficulties and keep in tune with the exciting new times with ease. The Issue : The annual time frame for computers and software were designed with the last two digits up to the year 1999 Ñ 99 for 1999. When 1999 is over, the computers according to the old or existing software, will print 00 for the year 2000 but in computer it will actually be considered the year 1900, a full 100 years behind the real time. Unless corrected, that could be a near catastrophe. The question now is how is it being solved? It is now necessary that when the year of the date is mentioned all four digits should be printed. This is being included in all new software's. The cost of doing this all over the world runs into tens and hundreds of billions of dollars. During the early period the world sensed that airplane flights would be affected in midair and on the ground because today's jet planes and other aircraft are highly sophisticated and computerized. It was feared that health records, especially in the advanced countries, could go haywire and the patients' treatment would be affected. In India, airline and railway ticketing could have been affected as most of it, if not all of it, is computerized. The approach to the solution : Recognizing this and sensing of other problems which would arrive,

an inexorable approach to meet the Y2K challenge has been launched. In India a task force has been set up with a Member of the Planning Commission at its head to monitor the progress and compliance with norms set to meet the needs. India is taking precautions for a few thousand years in the systems now being programmed. After 1999 the software should read 2000. After 2099 it should be 2100. After 2999 it should be 3000 and go up to the year 9999 smoothly. It has been visualized that after 31. 12. 9999 it will be 1. 1. 10000 the next day. That is eight thousand years away - far away indeed -and the resolution of the problem can be safely left to the scientists at that time. It is realized that several Indian banks have gone in for complex computerization programs and they must spare no effort to comply with the Y2K norms before the end of the year as they play a crucial role in the economy. The Reserve Bank is monitoring the banks' efficiency in this area. Several specialized institutions in the public and private sectors are providing facilities to the people and organizations to have the new equipment they buy tested to ensure that they meet the Y2K requirements. Communications and telephones are increasing in number very rapidly. India has now added more than 1 million new lines and twenty-one million direct lines.

4. Current Issue : Y10K -A Deca Millennium Bug

4. 1. Description : Much attention has been paid to the so-called " Y2K" problem as we approach the end of the millenium. The programs wrote by the programmers were designed to fail in the year 2000. The current fixes could lead to the crisis in the year 10, 000, when the programs may fail

again. This paper provides the brief description of the Y10K problem and solutions to the " Y10K" problem which has also been called the " YAK" problem (hex) and the " YXK" problem (Roman numerals). This specification provides a solution to the " Y10K" problem which has also been called the " YAK" problem (hex) and the " YXK" problem(Roman numerals). Various approaches exist for formatting dates and times. All these approaches have limitations. The 2-digit year runs into trouble next year. The 4-digit year runs into trouble in the year 10, 000. A 16-bit year runs out in the year 65, 536. A 32-bit counter for the number of seconds since 1970 [UNIX] wraps in 2038. In this specification, the main emphasis is on the upcoming Y10K problems since they are most common and a large number of existing standards and protocols are susceptible to them. These standards, and new proposals, will lead to a serious world-wide problem unless efforts are made now to correct the computing, government, and business communities.

4. 2. Issues of Y10K

There are various issues related to Y10K. Some of them are:.

- 1) What cannot be directly compared?? Dates from different calendar systems. For example, dates from the Aztec, Buddhist, Jewish, Muslim, and Hittite calendars must be converted to a common calendar before comparisons are possible.
- 2) Re-numbering of years is not covered for the future years. If, and when, a new " Year 0" occurs and comes into general use, old dates will have to be adjusted.
- 3)Continued existence of Earth-centric time periods (year, day, etc.) are problematical past the up-coming destruction of the solar system (5-10 billion years or so). The use of atomic-time helps some since leap seconds are no longer an issue.
- 4)Future standards and methods of synchronization for inter- planetary and inter-galactic time have not been

agreed to. 5) Survivability of dates past the end of the universe is uncertain.

4. 3. Structure of Y10K Solution.

4. 3. 1. Compatibility

The format should be compatible with existing 4-digit date formats. Y2K compliant programs and standards must continue to work with Y10K dates before the year 10, 000. Y10K compliant programs can gradually be developed over time and coexist with non-Y10K compliant programs.

4. 3. 2. Simplicity and Efficiency

Y10K dates must allow dates after 10, 000 to be easily identified. Within a program, there must be a simple procedure for recognizing the Y10K dates and distinguishing them from legacy dates.

4. 3. 3. Lexical Sorting

Y10K dates must be sortable lexically based on their ASCII representation. The dates must not require specialized libraries or procedures.

4. 3. 4. Future Extensibility

Y10K dates must support arbitrary precision dates, and should support dates extending arbitrarily far into the future and past. Y10K dates from different eras and with different precisions must be directly comparable and sortable.

4. 3. 5. Syntax Overview

The syntax of Y10K dates consists of simple, printable ASCII characters. The syntax and the characters are chosen to support a simple lexical sort order for dates represented in Y10K format. All Y10K dates must conform to the following rules. The syntax of Y10K dates consists of simple and printable ASCII characters. The dates are represented in the lexical sort order for the Y10K format. Every 10K date must begin with a Y10K year. Following the year, there may be an arbitrary sequence of digits. The digits are interpreted as MMDDHHMMSSmmmmuuunnnpppfff... (month, day, hour, minute, second, milli-second, micro-second, nano-second pico-second, femto-second, etc. - moving left to right in the date, digits always decrease in significance). When

comparing dates, a date precedes every other date for which it is a prefix. So, the date "19990401000000" precedes the date "199904010000000000". All dates and times must be relative to International Atomic Time (TAI) [NRAO]. In particular, dates with the format YYYYMMDD are interpreted to represent the exact instant that the day begins and precede any other date contained in that day.

1. Years 1 - 9999
The syntax of the current 4-digit year covers all years from 1000 - 9999. These years are represented as 4 decimal digits. To bring the year to 4 digits leading 0's must be added to the years before 1000. Files containing legacy pre-Y1K dates will have to be converted.
2. Years 10,000 through 99,999
It is not sufficient to represent dates beyond the year 9999 using the 4-digits. So, all years from 10,000 - 99,999 are represented by 5 digits preceded by the letter 'A'. So, 10,000 becomes "A10000" and 88888 becomes "A88888". Since 'A' follows '9' in the ASCII ordering, all dates with 5-digit years will follow all dates with 4-digit years (for example, "A10000" will sort after "9999"). This gives us the sort and comparison behavior we want.
3. Years 100,000 up to 10³⁰
By a simple generalization seen above, 6-digit years are preceded by the letter 'B', 7-digit years by 'C', etc. We can cover all years up to 10³⁰ (the last year representable is "Z99999999999999999999999999999999") using just 26 upper-case ASCII characters. Since the ASCII characters are sorted alphabetically, all dates are sorted appropriately.

4. Solutions to Y10K problem
One proposal [Wilborne] directly deals with the Y10K problem. In that proposal, dates are represented as decimal numbers with the dates compared numerically. The proposed format is simply YYYYMMDD - i. e. 5-digit years. To allow numerical comparison of dates, this representation

requires a completely fixed representation for the date. There can be no optional fields, the date resolution is limited to the granularity of one day, and this solution fails in the year 100, 000 (Y100K). The proposal is that the dates are represented as decimal numbers with the dates compared numerically. The proposed format is simply YYYYMMDD - i. e. 5-digit years.

5. Conclusion

Y2k is a serious problem, especially in the developing world. We need to seriously deal with it both personally and in our organization. Most personal computers and software will not be affected. The larger affected systems are mostly being worked on. Yes, computer errors will cause serious problems in some local areas. Many faxes will have the wrong date; some databases and spreadsheets will have wrong calculations until they are fixed. Some medical systems will have to rely on manual rather than automated measurements for a time. The intelligence and creativity of man will once again allow us to cooperatively solve issues that surface in the early hours of January 1, 2000. Spending money on a problem of a rather obscure nature and uncertain consequences shows the importance of technology issues for developed digital economies. The attention paid to Y2K and the amount of money spent by organizations in both public and private sectors is a good indicator of the level of development of the digital economy in a country.