Y2K - A NEW MILLENNIUM OF SUBROGATION AND RECOVERY CLAIMS

RICHARD C. BENNETT, ESQUIRE
COZEN AND O'CONNOR
1900 Market Street
Philadelphia, PA 19103
(215) 665-2000
rbennett@cozen.com

Atlanta, GA
Charlotte, NC
Cherry Hill, NJ
Chicago, IL
Columbia, SC
Dallas, TX
Los Angeles, CA
New York, NY
Newark, NJ
Philadelphia, PA
San Diego, CA
Seattle, WA
W. Conshohocken, PA
Westmont, NJ

I. FACTS

A. The Y2K Problem

The legal issue here involves insurance coverage for what is variously known as the "Year 2000," "Y2K," or "Millennium" bug or virus problem. This is a potential time bomb that may shut down many of the world's computers or cause them to act erratically on or before January 1, 2000.[1] As a starting point, therefore, it is important to understand exactly what the Y2K virus is and how it can be fixed.

When computers first came into being in the early 1960's, computer memory was a relatively expensive commodity. As a result, most computer languages and the software applications that employed those languages used a 6-digit format for inputting dates. This is frequently abbreviated as M1vVDD/YY; two digits and only two digits were used to input the month (M), the day (D), and the year (Y). The software automatically "completed" the information on what year it was by assuming the date in question was a 20th century date. The year 1967 was input by a human being and stored by the computer as "67," for example, and this meant that only two characters -- two "bytes" of computer memory -- were necessary to store the year instead of four.

The problem is that such software was not written with the future in mind. An application that automatically assumes that each year is a 20th century date is unable to correctly recognize any date after December 31, 1999, and any date-sensitive transactions or calculations

[1] It is erroneously associated with the Millennium, for that actually begins on January 1, 2001.

involving a2lst century date are impossible to carry out correctly. The software typically assumes that a "00" year entry means 1900, for example.[2]

Software can react to this situation in one of three ways. Assume, for example, that the calculation in question involves determining the difference between the present date and an employee's termination date of 1995. On January 1, 2000, an application that reads that date as January 1, 1900 may: (a) compute and record the difference as 95 years; (b) compute and record the difference as -95 years; or (c) stop functioning altogether - "crash" in computer terminology -- and send an "error" message to the operator.[3]

B. <u>The Fix</u>

The "fix" involves two things. First, all data stored on a company's hard drive must be changed from a two-digit format to a four-digit format. This is a relatively straightforward and comparatively inexpensive process, for such data is typically easy to recognize. Indeed, in most cases, a computer program can be set up that will do the lion's share of this automatically.

The other aspect is far more difficult and time-consuming, however, for it involves modifying all of the company's software programs or applications in such a manner to make them Y2K "compliant." A single program may consist of hundreds of thousands of lines of "source code," which is software code that can be read and modified by a human programmer. Each and every line of this source code must be examined, and any line containing a date-

---

[2] This is the rule with larger, mainframe computers. Most personal computer ("PC") software is designed to default to 1980 instead of 1900. A typical PC software program would, therefore, read a date entry of "04" as 1984 rather am 1904.

[3] The problem is compounded by the fact that the year 2000 is a leap year while the year 1900 was not A year divisible by 100 is not a leap year unless it is also divisible by 400. As a result, the date entry 02/29/00 is unrecognizable to a program that treats "00" as 1900; there was no leap year's day in 1900.

sensitive transaction or calculation must be changed to insure that the program is now using a four-digit format for the year. In addition, the programmer must also make sure that any such changes do not effect the functioning of the software, which is the time-consuming aspect of the task and the one that requires a degree of programming skill.

The process of modifying a company's software to make it Y2K compliant is a lengthy one. First, the company must identify which computer systems and their attendant software might need to be modified. Second, the problem must be analyzed and corrected by imputing new data and examining and correcting each line of existing source code as necessary. Computer systems may have to be taken down or off-line in order to do this. Third and finally, the fix must then be tested and "debugged" to insure that it works.

As might be imagined, this is a costly and labor-intensive process. Estimates of the cost range from approximately $1.00 to as much as $8.50 per line of source code (Inside Lawyer, November, 1996; Computer Lawyer, December, 1996). As of late 1996, one commentator reports that the Department of Defense had over 358 million lines of source code that need to be examined, and the cost of correcting this could range as high as $3 billion. Federal Express is reported to have become Y2K compliant in 1997, at a cost of $500 million; this works out to fully $5.00 per line of code (Boston Bar Journal, May-June, 1997).

The most commonly quoted figure in the literature for the total worldwide cost of this effort is between $300 and $600 billion (Forbes, July 28, 1997; Mealey's Litigation Reports, June 25, 1997; ABA Journal, June 1997).

The problem is principally one that affects large mainframe computers. PC's and microcomputers did not even come into existence until the early 1980's, and most software written for such platforms is now Y2K compliant and has been for some time. PC's do have internal clock systems that will need to be reset, and most also employ date-sensitive utility

routines, but these are easy to modify. As a result, businesses that did not computerize until after 1980 - such as the typical law firm -- will not be the primary target of the virus.

Businesses that did computerize early on, however, such as banks and insurance companies, will suffer the brunt of any problem. Mainframe computers are a substantial investment, and most institutions that have purchased them intend to continue to employ such platforms until well after January 1, 2000. Because of the advent and present dominance of the PC, many mainframe installations now operate by using what is called "legacy" software -- applications developed years ago that are still in use even though: (a) the original vendor is frequently out-of-business; (b) there may be little or no source code documentation left to assist in effecting a fix; and (c) the program employs an obsolete computer language such as Cobol, RPG, PL/l, Jovial or Fortran. Legacy applications still control the principal databases maintained by large corporations such as banks, and they will be particularly difficult to make Y2K compliant.

C. Knowledge of the Problem

There is a clear consensus in the literature that knowledge of the Y2K problem is virtually universal. See, e.g., Computer Lawyer, December, 1986 ("The year 2000 problem has been well known for years and is completely within the control of the insured to correct [T'); Inside Lawyer , November, 1996 (The problem is so well known that "ignorance cannot be a defense" any more). Indeed, there are even a number of internet web sites devoted exclusively to this topic. See, e.g., http://www.year2000.com, the site maintained by Peter de Jager, a recognized expert in the field.

In spite of this, however, commentators estimate that there will be many businesses that will not be fully Y2K compliant before the Millennium is upon us. According to Forbes, for example, half of the companies with large computer systems simply won't get the job

done in time. (Forbes July 28, 1997; see also Mealey's Litigation Reports, June 25, 1997; ABA Journal June 1997).

This is, in part, because many businesses apparently believe that someone will ultimately develop a so-called "silver bullet" - a brilliant software solution that will allow computers to fix themselves easily and cheaply, obviating the need for expensive and time-consuming human examination of programs on a line-by-line basis. No one has done so thus far, however, despite the depth and breadth of talent that is presently working on this problem, and the longer a business waits, the more difficult it becomes to "get the job done" by January 1, 2000. The fact that many large companies' problems involve unique mainframe legacy applications also means that such a universal solution is extremely unlikely to be devised.

All firms worldwide will undoubtedly back-up their computer systems on December 31, 1999, but all this does is to give the company the ability to restore programs and data after their computers crash the next morning. The computer systems themselves must still be made Y2K compliant or they will simply crash once again after having been restored from such a back-up.

D. Existing Problems

Y2K problems are already surfacing around the world, and this will snowball as the Millennium approaches. One reason is that this is a "Year 1999" problem for many systems, for a good deal of software uses the year entry "99" to provide special instructions to the computer. Tape library management systems, for example, frequently use such an entry to direct the computer to either destroy a particular tape library or to render it inactive.

In addition, the nearer we get to January 1, 2000, the more problems will surface. A Portland, Maine insurance company began experiencing difficulties with the Y2K virus in 1995, for example, when its computer began deleting active files. The software program had

been instructed to remove dormant policies from the system if there was no activity in connection with those contracts of insurance for a period of five years or more. It accomplished this by retrieving the date of the last activity and then adding five years to this; if the resulting date was prior to the current one, then the policy was deleted.

The system performed without a problem until December 31, 1994. Where the last activity was undertaken in 1995, however, the software added five to "95," obtained a "00" result, read this to be the year 1900, and the proceeded to delete the policy as a result. The problem was not caught until early 1996, and by that time the computer had deleted literally thousands of policies from the system (ABA Journal, June 1997).

In addition, the literature that we reviewed contains at least two reports of existing litigation. According to the August 18, 1997 edition of The Financial Times, a Detroit supermarket owner recently filed suit against Tec-America, a cash register supplier based in Atlanta, after cash registers supplied by the defendant refused to accept and honor any credit cards with expiration dates of 2001 or beyond. Plaintiff sought to recover losses in excess of $100,000, alleging that this had occurred some 150 times over the course of a two year period.

In addition, the July 28, 1997 issue of Forbes reports that there was recently another lawsuit which was settled under seal. This involved claims by an international magazine publisher against an outside computer vendor, and it was allegedly settled for a figure in excess of $4 million.

E. Failure Scenarios

A computer stores, reads, and manipulates data using a binary system of ones and zeros. A single character, such as "9" or "5," takes up one byte of storage space. To the computer, that byte is an 8-bit string of ones and zeros; 9 is 0000 100 1, for example, while 5 is 00000101.

Most businesses today still employ magnetic storage media such as hard drives to store the information necessary for day-to-day operations. These are then backed-up using either magnetic tape or optical disks. To store a number such as "9" on its hard drive, a computer's software instructs the drive's head to move to that portion of the disk where the character will be stored and then to "write" the character to a series of eight "bit cels." The head induces magnetism in the surface material of the drive's disk, and it magnetizes bit eels in one direction to represent binary ones and in the other to represent binary zeros. The drive remains polarized after the machine is switched off, and the information is thus permanently stored until replaced.[4]

Large businesses are increasingly using read/write optical disks in lieu of magnetic storage media for the day-to-day storage of information. These use a laser to "etch" the surface in order to record information, but the important point is that, like their magnetic brethren, they retain all of the data that has been "written" to them until it is overwritten or replaced.

There are two paradigmatic failure scenarios -- computers will either make incorrect calculations where dates are involved or they will not work at all. As discussed above, for example, a computer instructed to determine the difference between an employee's termination date of 1995 and the present time (as of January 1, 2000) might report: (a) 95, (b) -95, or (c) "error." All three possible results leave the date stored in the computer's magnetic or

[4] "Certain systems are less vulnerable to the problem because they store dates in a different fashion. In the example above, it is assumed that the date is stored as a representation of the characters for "9" and "5." Most Unix Systems store dates differently, however, for they use what is known as "reference dating." Instead of storing an employee's termination date as June 30, 1995, for example such a computer would first determine the number of days that have elapsed since a particular "reference date;" in many Unix Systems, the reference date is November 1, 1982. Such a system would then store the termination date as the number of days that have elapsed since the reference date rather than storing it as the calendar date of June 30, 1995. In this case, for example, the date would be stored as "day 4625." A computer using reference dating is less susceptible to the Y2K problem.

optical storage media unchanged, however; nothing has happened to the stored termination date of "95." The computer program has simply either used that date to make an incorrect determination of how long ago the employee left or it has performed the calculation, rejected the result, and shut itself down as a result, flashing an "error" message to alert the operator to the fact that something is amiss. In neither case, however, has there been physical loss or damage to the machine or to the "95" date that it has stored away.[5]

The problem is, of course, that each of the archetypical failure scenarios can lead to many other types of loss or damage. A computer that makes incorrect calculations may then start to cause other types of loss to its own system. The Portland, Maine insurance company's computer, for example, did incorrect calculations that caused it to delete files off of its magnetic storage media. This clearly constitutes loss of the data that those files represent.[6] Incorrect calculations can also lead to a business interruption loss occasioned by the need to shut down and restore any deleted files. A computer that crashes will obviously also occasion a business

---

[5] While one might conclude, in everyday terms, that no physical loss has taken place, that is not necessarily determinative as to whether such a "loss" under a first-party property insurance policy is compensable.

[6] Even this is subject to at least one caveat, however, and that illustrates just how fact-sensitive and complex this issue is. The typical PC stores a file by setting aside the necessary sectors on its hard drive and then polarizing the surface of the drive in order to "write" data to it. Each hard drive then has one or more file allocation tables (TATs") which are used by the software to determine where that particular file has been stored on the drive.
When a file is deleted, those sectors on the hard drive that contain the file itself are usually left unchanged. That is to say that they remain polarized and still contain the information itself All that the computer does is to modify the FAT to reflect the fact that these sectors are now "free" and can be used to store new files on the drive. The actual data in these sectors is not "erased" until a program uses the drive's head to "write over" those sectors by storing a new file there. It is only after this happens that the data has actually been eliminated from the surface of the drive.

interruption loss. Finally, a crash can also lead to direct physical loss or damage to one or more of the system's hard drives. If a computer "locks up" and has to be turned off without first closing out all of the applications that are running and then initiating an orderly shutdown, the disk - which is spinning at 3600 RPM or more -- may come into contact with the drive head, resulting in a literal head crash that requires replacement of the entire drive.[7]

Incorrect calculations may themselves be stored as invalid or corrupted data, and the use of such invalid or corrupted data in additional calculations will cause the problem to mushroom. Finally, both incorrect calculations and a crash may lead to some other form of property damage or business interruption loss to the insured and/or to third-parties that is not limited to the computer itself.

The very novelty of this problem and the pervasive use of computers in connection with literally every facet of our lives makes it impossible to foresee anything but a tiny percentage of the possible failure scenarios, but at least some come readily to mind.

The prototypical first-party claim will be a business interruption loss occasioned by virtue of the fact that a company's computers are either off-line or making incorrect calculations, leading to corrupted or invalid data that must be searched for and corrected.[8]

Thus, banks may find that they cannot print or deposit checks, properly calculate interest for depositors' accounts, process electronic transfers of funds, amortize loans, or determine when loan payments are overdue. Premium payments to insurance companies may be processed incorrectly or rejected as stale, and the computer may issue improper cancellations or non-renewals to policyholders. In addition, carriers may find that they cannot properly calculate

---

[7] This is, indeed, what "crashing" a computer originally meant.

[8] As a perusal the following indicates, incorrect data or shutdown computers can also give rise to a number of third-party liability scenarios.

annuity benefits for policyholders, and they may derive incorrect results when their computers consult actuarial tables.

Retailers' computers used to monitor inventory and "sell by" dates may deem fresh products to be dated or subject to condemnation. Indeed, even the phone company may find that it cannot place or route calls.[9]

There is also the potential for substantial first-party property damages losses. As an example, a date-sensitive calculation improperly performed by the computer that is monitoring a drill rig inside a potash mine might cause the rig to continue drilling into a water-bearing aquifer that overlays the mine when it should instead shut down, causing the mine to flood out. By the same token, a computer responsible for opening and closing the valves in a slurry line at a plant that extracts crude oil from tar sands could malfunction because it misreads the date, causing an explosion and fire and damaging or destroying the entire facility. Similar examples are legion.

Finally, first-party claims will also result from the activity of third-party suppliers and/or customers of one's insureds. As one commentator has observed, "no software program is an island" (Mealey's Litigation Reports June 25, 1997). Businesses generally depend on the exchange or sharing of information from a number of outside parties. Like any computer virus, Y2K can reinfect a system that has been made compliant if suppliers' or customers' computers are feeding incorrect information to it. This is a two-fold problem; businesses sending or receiving electronic data from others must both insure that those on the other end of the line have been made Y2K compliant and insure that they have done so in a way that is compatible with their own computer software solutions.

---

[9] A computer that treats a "00" as 1900 might bill a customer who began a call at 11:59 p.m. on December 31, 1999 and completed that call two minutes later for 53,000,000 minutes of time!

Third-party claims come in equally assorted forms. Thus all companies may face the possibility of liability for failure to disclose the existence of Y2K problems and their potential to interrupt business operations on SEC filings such as Form I0K's and I0Q's if they have reason to believe that they are unlikely to be compliant in time. Failure to do a proper due diligence may also subject corporate officers and directors to liability if their company acquires a firm beset with Y2K problems. In addition, failure to take appropriate steps to erect "firewalls" in order to prevent the computers of third- parties such as customers and suppliers from reinfecting and thereby corrupting one's own Y2K compliant systems may be the source of such claims.

Pension funds and the like may find that computers improperly terminate benefits because date comparisons show that retired employees haven't even been hired yet. Benefits packages are inherently date-sensitive, and disability benefits and the like typically hinge upon the date of the onset of the disability and the hire date of the employee at issue. Retirement benefits are similarly pegged to the date of birth and the date of hire as well as the date of retirement. Incorrect benefit calculations will undoubtedly result in claims against the employer and the fund.

Accountants could face liability if they provide incorrect profit projections based on corrupted data. Professional liability claims may also be lodged against Y2K consultants and companies' own in-house information systems departments where existing software has been modified by either in order to make it Y2K compliant. Such modifications may well be deemed a derivative work and thereby give rise to a cause of action for copyright infringement by the software's designers.

Finally, there is also significant potential for property damage or business interruption claims against product manufacturers. The potash mine and the synthetic crude oil

plant discussed above would clearly have a cause of action against any third-party entities that designed, sold, or serviced the monitoring computers that malfunctioned, and similar examples abound. Thus a hospital's computer system may fail as a result of the Y2K virus, precluding access to critical patient medical records or causing incorrect medication to be administered to those in the facility's care.

In large part, such claims will also arise from the fact that virtually every product in today's marketplace uses some sort of "embedded" microprocessor, which is to say a small computer that is dedicated to a single function and is not reprogramable. The simplest example is a wristwatch; even the most inexpensive digital watch has a microprocessor that displays the day of the week and the date. A watch that is not Y2K compliant will show January 1, 2000 as a Monday (January 1, 1900) instead of a Saturday.

This is obviously a trivial example, but more serious ones are easy to conjure up.

A hospital, for example, may have time- and date-sensitive devices that monitor the telemetry from sensors on patients' hearts and other organs. Traffic lights with embedded microprocessors may fail to keep track of the date and time, leading to personal injury and property damage occasioned by auto accidents. Bank vaults with embedded microprocessors may fail to open, and the components of security systems may malfunction.

The important point is that we are dealing with a continuum of possibilities in every instance. In the first-party property damage context, for example, one end of the spectrum is represented by the computer that merely stops functioning, having done no damage to itself or to anything else and having triggered no insurance obligation as a result. At the other end of the spectrum lies the computer which stops monitoring and controlling the nuclear reactor of an atomic power plant on midnight on December 31, 1999 and thereby permits a catastrophe to occur. The question -- which is virtually impossible to address in a vacuum -- is where along

that continuum each type of loss might lie, and whether the critical triggering event (in this case, direct physical loss of or damage to property) has taken place.

II. INSURANCE COVERAGE ISSUES

The very novelty of the problem and the pervasiveness of computer systems makes it impossible to foresee anything but a small percentage of the probable types of insurance claims, but our preliminary conclusions with respect to coverage under some of the many insurance forms with which we are familiar are as follows:

A. First-Party coverages -Fortuity

Under the first-party forms, we believe that most types of property damage that will arise are excluded. The reason for this conclusion lies not in the language of the forms themselves, however, for these were not written with Y2K in mind, and they arguably afford coverage for many foreseeable claims. It is, rather, the non-fortuitous nature of most losses that will operate to bar coverage.

Even all-risk policies afford no coverage for loss that is certain to occur. To be compensable, the loss must be "fortuitous," which is to say that it must result from a risk. This implied requirement is universally recognized.

At first blush, this would appear to be a potent defense to My first-party Y2K claims, but the fact of the matter is that it is the Millennium that is certain to occur -- a Y2K loss is not. A loss is only non-fortuitous if. (a) it is within the insured's knowledge and control to avert that loss; and (b) the insured has failed to take reasonable precautions in an effort to do so.

Here, the solution to the Y2K problem is known, and the implementation of that solution is within each insured's control. As a result, insureds who make little or no effort to prepare for the Millennium will not be able to show that they have sustained a fortuitous loss. In

addition, most insureds who suffer loss because they have either waited too long to start the process of effecting a fix or proven unwilling to allocate the necessary manpower and money to get the job done in time should also be barred from recovery. Those first-party insureds who have done everything within their control will not be precluded from recovering on the basis of the fortuity doctrine merely because a program or a section of source code has been missed, however.[10]

B. First-Party Coverages - Policy Language

Non-coverage arguments bottomed on the actual language of most forms are more problematic. The term "personal property" is frequently undefined, and electronically-stored data is often specifically referenced as a type of personal property. Covered personal property also typically need not be "tangible" in nature, and an irretrievable loss or corruption of such data would probably be deemed to be a loss of covered property as a result.

Insuring agreements usually limit coverage to "direct physical loss of or damage to" insured property, but there is no caselaw that addresses the issue of whether this requirement has been satisfied where data is permanently lost or corrupted. In addition, coverage has been found by courts under similar language in situations in which the insured property, while undamaged itself, is rendered inaccessible due to landslide or fire damage to surrounding structures.

---

[10] There is one significant caveat to this conclusion. To a degree, courts' receptiveness to arguments that particular insureds' losses are non-fortuitous in nature will be directly proportional to the degree to which the insurance carrier itself succeeds in becoming Y2K compliant. The first question asked by counsel for an insured whose claim has been denied because insufficient resources were allocated to the problem will be whether the insurer experienced any major problems of its own. If the insurance carrier itself suffers a major loss, then it comes to court with "unclean hands" when it argues that such losses are non-fortuitous because they he within the insureds' control.

If the insurance carrier suffers a major loss and makes an insurance claim for i then any fortuity defense that it has is fatally weakened.

Finally, present day first-party forms' exclusions were not written with Y2K in mind, and none would appear to bar coverage for the typical kinds of losses that might be occasioned by the so-called Millennium bug. Specifically:

- "Hidden or latent defect" has been held to be limited to problems that could not be discovered by any known or customary test.
- "Artificially generated electric current" has only been applied in instances of electrical arcing, which is to say in situations where an electrical current has escaped from and is traveling outside of its normal distribution pathways.
- "Mechanical breakdown" refers to the failure of the working mechanism or moving parts of a device. A Y2K loss will be occasioned by computational errors by the computers' software algorithms.
- "Errors in processing" has only been applied by courts in cases involving complex chemical manufacturing operations, and the exclusion itself specifically references the manufacturing of the insured's products.
- "Acts or decisions, including the failure to act or decide" has been the subject of only one case that we discovered, and that decision held that any reading which applied this language to negligent acts that cause a loss "would leave the insurance policy practically worthless."
- "Faulty workmanship or design" is aimed at construction problems, and all of the decisions that have construed this exclusionary language have done so in such a context.

C. First-Party Coverages - Sue and Labor Issues

First-party forms usually contain a "sue and labor" clause, and such provisions impose an affirmative obligation on the insured to take all reasonable steps to avert or minimize a loss for which the carrier would otherwise be liable.

The obligation to sue and labor only comes into being after a covered loss has begun to occur, but it is our prediction that most of our insureds will have begun to experience some form of Y2K-related loss or damage well before the Millennium dawns, and they may then invoke such provisions in an effort to coerce carriers into providing financial aid for a fix. Their argument will be that this loss is certain to occur unless appropriate steps are taken, and it may

also be argued that any refusal by the carrier to assist in funding such efforts effectively undercuts any fortuity defense down the road because it deprives the insured of the wherewithal to get the job done in time. As a result, sue and labor is clearly an area in which additional research in connection with the ramifications of this problem is required.

D. Third-Party Coverages - The Insuring Agreement

As with first-party forms, the language of third-party coverage was not written with Y2K in mind, and there are, therefore, certain types of claims that arguably fall within the scope of coverage afforded by such contracts of insurance. In our judgment, however, many such claims should be excluded for substantially the same reasons as their first-party cousins.

In order to qualify as "property damage," the claimed loss must involve either physical injury to or loss of use of "tangible" property. Courts have typically held that such a term connotes property that is apparent to the senses or capable of being touched or felt. Thus, for example, the loss of benefits, wages, and salaries do not constitute loss of "tangible" property. Unfortunately, however, the caselaw addressing the question of whether electronically-stored data falls into this category is considerably less conclusive; most recent decisions appear to hold that while information itself is not "tangible" in nature, it becomes so once it has been "put on paper" or otherwise saved or stored in some fashion.

In our judgement, however, there will still be no coverage where liability claims are themselves occasioned by the insured's failure to take those steps necessary to become Y2K compliant. Covered bodily injury or property damage must itself be caused by an "occurrence," which is to say an accident that was neither expected nor intended from the standpoint of the insured. As with first-party claims, the insured cannot ignore a known problem such as the Y2K bug and then ask its carrier to reimburse it for any and all liability occasioned by its own

inaction. However, insureds who have taken all reasonable steps within their control will not be barred by the "expected or intended" requirement. [11]

E. Third-Party Coverages - Trigger

Under a liability policy, coverage is "triggered" when the bodily injury or property damage at issue has occurred, and caselaw has provided a bewildering array of answers to this seemingly simple question. Even a single jurisdiction may employ several different theories of trigger depending on the precise nature of the injury in question. Generally, the various theories of trigger and their implications for the Y2K problem are as follows:

- Exposure, which was initially applied in asbestos cases and defined as the time when the claimant was initially exposed to asbestos fibers. At least one recent Seventh Circuit case has held that the date when a defective plumbing system was installed in a home constituted the date of physical injury under this theory. A court applying an exposure trigger could hold that policies in place years ago when software unable to recognize a 21st Century date was originally installed in the computer system have been triggered by Y2K-related damage.

- Manifestation, which provides that coverage in place when loss first becomes manifest or known is triggered. The implications of such a trigger are uncertain, for arguably the injury that will be occasioned by Y2K is already known to virtually the whole world.

- Continuous trigger, which treats injury -- typically injury occasioned by a progressive disease - as occurring continuously from initial exposure to manifestation. Such a theory should only be applied in cases in which it can be shown that non-compliant software has been slowly and insidiously performing incorrect calculations and storing them in the form of incorrect or corrupted data over a period of years, but the fact that this theory usually triggers a number of policies has made it very attractive to many courts.

- Double trigger, which holds that both exposure and manifestation are "triggering" events.

---

[11] Occurrence" language is also germane to the question of how many limits of liability are available. Under the prevailing "cause" test, courts should hold that all loss due to a failure to become Y2K compliant constitutes a single occurrence.

- Injury-in-fact which basically rejects any generalized rule in favor of a case-by-case inquiry in an effort to determine when bodily injury or property damage actually occurred.

Given the fact that most insureds will almost certainly begin experiencing at least some problems prior to January 1, 2000, it is our (reluctant) conclusion that many courts may opt for a continuous trigger theory. The Y2K problem has the potential to be a massive drain on the economy, and many tribunals will perceive their task to be, at least in part, to insure the mobilization of all resources available in order to spread the burden of loss as widely as possible.

F. Third-Party coverages - Exclusions

Unlike their first-party relatives, third-party forms' exclusions will apply to preclude coverage for many types of Y2K claims. The most important examples are:

- "Expected or intended injury," which may bar coverage where insureds make little or no effort to effect a fix.

- "Contractual liability," which bars breach of contract claims.

- The so-called "business risk" exclusions for property damage to the insured's "own product" or "own work." These have been held by at least one court to bar coverage for claims that the insured was liable because it sold a defective computer program that failed to perform as promised.

- "Failure to perform," which excludes claims for a third-party's property that is undamaged but nonetheless "impaired" because it incorporates defective components manufactured by the insured. Under a typical failure scenario, where the computer system at issue merely stops functioning but is otherwise undamaged, this exclusion may also bar coverage for the manufacturer of any non-compliant software that - caused the crash.

- "Sistership," which excludes loss caused by the need to recall and repair a whole line of products after one or more prove defective. This language should come into play in instances where the insured is a product manufacturer whose products use embedded chips that are susceptible to the Y2K problem. When the failure of one causes the recall of the rest, such a provision should operate to preclude coverage for the resulting repair work necessary to make the balance of the insured's product line Y2K compliant.

G. Directors and Officers Coverages

Most of the D&O policies that we have reviewed are claims-made forms, and they present few "long tail" exposure issues as a result. There are a number of other potentially troubling exposures, however.

- Some D&O forms contain no reference to a retroactive date, which is to say a date after which the "wrongful act" must occur in order for coverage to be triggered. Many activities that could arguably fall within the rubric of "wrongful acts" may have taken place many years ago. An obvious example is the selection of a non-Y2K compliant software program or application. These are not excluded under the language of such a form.

- Other D&O forms provide coverage to directors and officers for more than merely those exposures they have by virtue of their corporate positions.

- Finally, the definition of "wrongful act" is frequently broad enough to encompass "status" claims within its purview. In other words, claims brought against the company that include the directors and officers as defendants and allege nothing more against them than their status as such may qualify for coverage.

There are, however, typically at least two exclusions which should be broad enough to bar coverage for many Y2K-related D&O claims. These are:

- The exclusion for damages arising out of bodily injury or property damage. This language should be sufficient to bar coverage for any bodily injury claims, though its reach is somewhat narrower with respect to those occasioned by property damage. In connection with the latter, the usual exclusionary language encompasses only claims "for damage to or destruction of any tangible property." We read such language as potentially providing coverage for claims for consequential loss as a result of damage to tangible property so long as any such claims can be cast as ones resulting from poor judgment of the company's officers and directors. In addition, such an exclusion is ineffective to the extent that the loss of or corruption to data can be characterized as damage to property of an "intangible" nature.

- The exclusion for any circumstance known to the insured which might reasonably be expected to produce a claim will also operate to preclude coverage for many insureds in light of the virtually-universal state of knowledge concerning the Y2K problem.

H. The Advisability of Adding Exclusionary Language

There is relatively little caselaw on the admissibility of subsequent changes in policy language, but the decisions that do exist suggest that carriers will have at least an even

chance of convincing a court that the addition of a Y2K exclusion is not relevant to the issue of whether or not existing language affords coverage for such exposures. Some courts have found that policy wording changes are presumed to have been intended "to impose a different liability," but those decisions have uniformly involved situations in which the previous policy language had already been the subject of court interpretation. No court has thus far construed any insurance contract language in connection with this particular risk. In addition, courts have also recognized that there be many reasons that an insurer chooses to amend its policy language. Finally at least one court has analogized the revision of policy language to "subsequent remedial or precautionary" measures and found that such revisions are inadmissible as a result under California's equivalent to Rule 407 of the Federal Rules of Evidence.

Q:\DEFENSE\93394\001 RAMPMP0496
PHILA1\1143142\1 099994.000
11/30/1999 1:28 AM