I'm Bob Bemer, and President of BMR Software, a Texas corporation concerned only with fixing or ameliorating the Year 2000 problem.

My qualifications are as an early contributor to the COBOL language, now associated with this problem, having coined the word "COBOL", and created its Picture Clause facility, where data size and type are specified. If programmers had written "PIC 9999" instead of "PIC 99" for year values, we would not be concerned now.

I warned of this danger in 1979, in my article "Time and the Computer", which became the primary document on time in the Internet design.

The Escape Key on your keyboard, upper left, is a reminder that I invented the "Escape Sequence", which is how you change colors on the screen, switch from Roman to the Cyrillic or Japanese Kata Kana fonts, vary fonts and such on your laser printer, etc. My Arizona license plates say "ESC SEQ" and "ASCII", which is the international standard alphabet code I am generally acknowledged as having "fathered". Again, a major Web and Internet component.

With these and other contributions I am recognized in computer history. Failure in a new venture is the last thing I need. But when it appeared that current corrective methods for the Year 2000 problem were going to be insufficient, I recognized three causes:

1) The source code programs that produce what we now call "legacy" code are almost universally poorly documented and difficult to comprehend. We can't plead with the authors; they've largely died or moved to Philadelphia or such.

2) For many legacy programs now running, source code has just plain disappeared. Some estimate this loss as high as 40%. Losing 40% of what it takes this country to operate? If you think I'm wrong, ask the IRS.

3) With increasing reliance upon computers to even think for us (see your grocery store cash register), the Government and the people have been remiss in the writing of dates. The three methods have been "month-day-comma-year", "day-month-year" for the military and formerly Europe, and the international standard "year-month-day" now -- like the metric system, almost universal except for the United States. That "030405" in your database may be a date, but which value is the year? Which the month?

So the press may stop blaming the programmers, who were only trying to mechanize your own methods. "The enemy is us" is a Pogoism. The problem was not caused entirely by computers, and it can't be fixed entirely by computers, but we can try.

I reasoned that programs were still running because the computer could still understand them, In "object code", even if the people couldn't understand them, in source code such as COBOL. Also that computer programs themselves would be required. They're a lot faster than people.

The object code method I devised was decribed generally by columnist Thomas Petzinger in the Wall Street Journal of 1997 Jun 20 (note that ordering). One result was invitations to keynote several conferences, among them one for State Government next April, and one for the Federal Government this December 09.

Being also conference chairman, I'll hear a lot of status reports from various agencies, and I predict they won't speak of triumphs. It is in preparation for this keynote that I am here today. My abstract says:

> "Governments acted too late when a single assassination began World War I, and again too late when Hitler began World War II. Their main faults were complacency and disbelief in the danger. Now our lives and well-being are in danger from our own lack of foresight, for we have allowed the computer to usurp our thought and action processes. We cannot return to our old ways -- we've forgotten how, and it's too late. But we can mobilize nationwide, the authority for which may be found in our Constitution, even in the very Preamble."

The parallels to war are striking. In World War II I worked at Douglas Aircraft, not voluntarily, but by assignment. There was a War Production Board, and machinery for making lollipop sticks was turned to something for guns. Who owns a 1943 Ford? We all worked together against the common threat. We have such an enemy now – possible collapse of our living sytems.

Yes, the capitalists are correct. The free market works. But will it work to an inexorable deadline like the Year 2000? What will rally us to the common good, and even survival?

THE WHITE HOUSE
WASHINGTON

May 1, 1998

Mr. Robert W. Bemer
President, BMR Software, Inc.
1100 Centennial Boulevard, Suite 140
Richardson, TX 75081

Dear Mr. Bemer:

Thank you for your interest in helping to solve the Year 2000 problem.

The President's Council on Year 2000 Conversion, which I chair, is working with federal agencies as well as those outside the government, to resolve the "Y2K" issue. Specific offers of help from individuals are being accepted by the Social Security Administration, the U.S. Department of Commerce and the Small Business Administration. My assistant, Phyllis Kaiser-Dark will be happy to discuss this matter with you further. She can be reached at (202) 456-7171.

Best wishes.

Sincerely yours,

John A. Koskinen
Assistant to the President and
Chair, President's Council on
Year 2000 Conversion

cc: Sue J. Smith

GAO: Year 2000 Computer Problems Persist

By RAJIV CHANDRASEKARAN *and* STEPHEN BARR
Washington Post Staff Writers

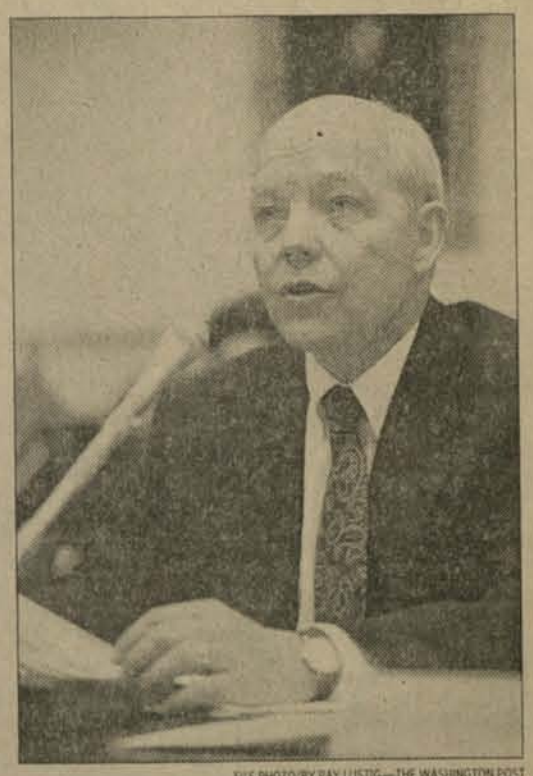
A congressional investigator warned yesterday that several federal agencies still are not moving fast enough in repairing their computer systems so they will work in the year 2000. Despite an expected last-minute push, he predicted that some agencies will not finish in time, increasing the possibility of botched financial transactions, grounded airplanes and inoperable military systems.

The warning—issued by Gene L. Dodaro, the General Accounting Office's assistant comptroller general—follows a recent Clinton administration report raising costs of the repair work by $800 million and indicating that only 35 percent of the government's "mission critical" computer systems have been fully repaired.

"At the current pace, it is clear that not all mission critical systems will be fixed in time," Dodaro told a joint hearing of two House technology subcommittees. The GAO has predicted for about a year that some crucial systems will miss the deadline, but yesterday's appraisal was delivered in more bleak tones and raised the possibility of government computer failures leading to spillover problems for the private sector. "America's infrastructures are a complex array of public and private enterprises with many interdependencies at all levels," he said.

The potential chaos stems from the fact that many computer systems use a two-digit dating system that assumes that 1 and 9 are the first two digits of the year. Without specialized reprogramming, the systems will recognize "00" not as 2000 but 1900, a glitch that could cause the computers either to stop working or to start generating erroneous data.

John A. Koskinen, a former Office of Management and Budget official who was tapped by President Clinton last month to spearhead the government's conversion efforts, agreed that agencies need to hasten their progress, but he urged the subcommittees not to advance doomsday scenarios.



FILE PHOTO/BY RAY LUSTIG—THE WASHINGTON POST

John A. Koskinen, who is spearheading the government's year 2000 conversion efforts, urged congressional subcommittees overseeing the issue not to advance doomsday scenarios.

"We need to avoid creating panic and precipitous, counterproductive activity," Koskinen said. Talking to reporters after the hearing, he said that agencies "are confident of meeting the March '99 deadline" established by OMB for systems to be repaired and tested.

> "Not all mission critical systems will be fixed in time."
>
> **Gene L. Dodaro,**
> GAO assistant comptroller general

Koskinen, who directs a presidential council on the year-2000 issue, said his group will work with state and local governments as well as the private sector to ensure that data-exchange activities will continue to function smoothly after repairs are made.

The OMB said last week that as of Feb. 15, about 35 percent of the 7,850 mission critical systems have been fixed. About 45 percent still have to be repaired, 15 percent will be replaced and 5 percent will be retired.

Mission critical systems include those that process benefit checks, crunch tax returns, monitor food safety, direct air traffic and support national security activities.

The OMB, which estimated the costs of fixing the problem at $2.3 billion last February, said the price tag has mushroomed to $4.7 billion, including an $800 million increase between November and February.

Although OMB officials have long said they expect the final figure to be about $5 billion, Rep. Stephen Horn (R-Calif.), who co-chaired yesterday's hearing, predicted the cost will increase substantially: "I think you're going to be close to $10 billion when you're done with this."

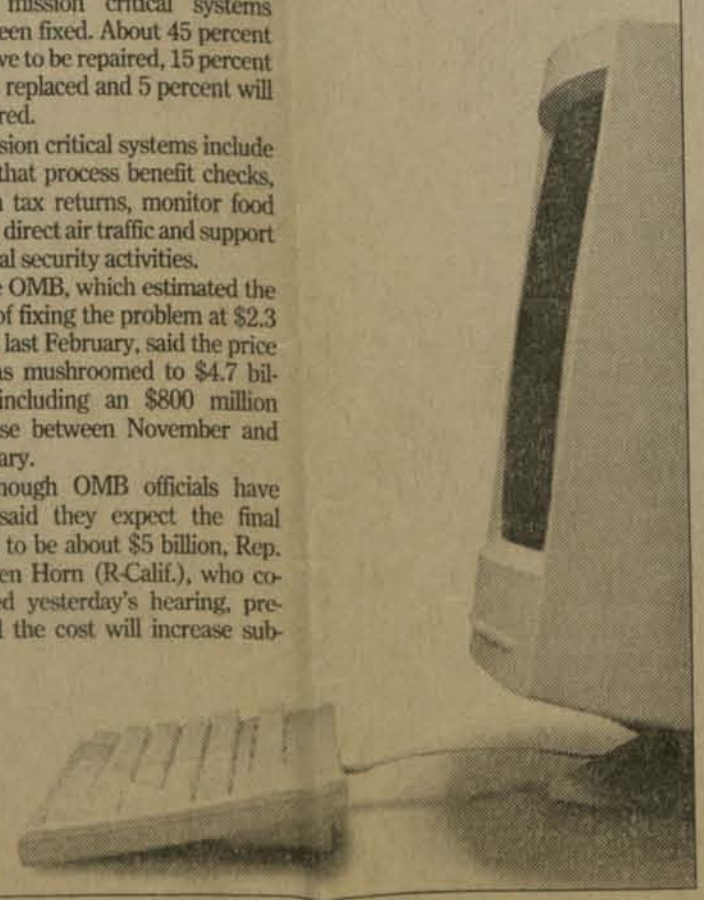
Horn recently released a report card on agencies' progress that flunked the departments of Education, Defense, Transportation, Labor and State. At their current rate of repairs, Horn said Education would finish fixing all its mission critical systems in 2002, Transportation in 2003, Labor in 2007, Defense in 2009 and State in 2014.

"We have a lot of work to do and little time remaining," Horn said.

Horn also raised concerns about plans to fix the 60,000 non-mission-critical systems in federal agencies.

"We must constantly remind ourselves that the mission-critical systems we talk about are but the tip of the iceberg," he said.

The hearing was held by the House Science subcommittee on technology and the Government Reform and Oversight subcommittee on government management, information and technology.



From Gore Task Force, 200 Ways to a Better IRS

By STEPHEN BARR
Washington Post Staff Writer

Vice President Gore yesterday issued a task force report making 200 recommendations to improve the Internal Revenue Service, saying that "when Americans sit down at their kitchen tables to fill out those tax forms, they have a right to know that their government is working for them and not against them."

But Republicans dismissed the report as nothing new and criticized Gore for belatedly joining the fight against IRS abuses.

"The congressional IRS reform train had already left the station, and today the administration hopped into the caboose," House Ways and Means Committee Chairman Bill Archer (R-Tex.) said.

Gore's White House announcement and the prompt GOP counterattack provided new examples of the heated political competition to take credit for reforming the IRS. The House approved a reform bill last year, and within the next two weeks, the Senate Finance Committee is expected to finish drafting its legislation to overhaul the agency.

Gore's recommendations, some of which have already been implemented, ranged from providing extended walk-in service at more than 150 IRS offices across the nation on the last six Saturdays of this year's filing season to a proposal authorizing IRS district directors to issue emergency tax refunds within 24 hours to citizens facing economic hardship.

Gore said new task force recommendations would make it easier for 3 million small-business taxpayers to file quarterly payroll returns by telephone, would create a "small business laboratory" in Seattle to help companies navigate tax procedures and would designate March 28, April 4 and April 11 as special days to help low-income earners apply for the Earned Income Tax Credit when preparing returns.

Page two of the Gore report also included a finding that probably reflects the feelings of many taxpayers: "What IRS customers say they want . . . minimum contact with the IRS."

The report, "Reinventing Service at the IRS," relied on the insights of an IRS task force that included 30 "front-line" employees who work in field offices. Members of the task force attended the Roosevelt Room event, where Treasury Secretary Robert E. Rubin and IRS Commissioner Charles O. Rossotti also spoke.

"We are not just tinkering at the margins here. We are effecting fundamental change," Rossotti said.

National Treasury Employees Union president Robert M. Tobias welcomed the report, calling it an opportunity for the IRS, "working with its employees, to shape a new future."

Task force member Marilynn Smith of the Kansas City IRS Service Center told Gore that "my co-workers in Missouri can already see the changes in the way we do business. And believe me, if they see it, that's saying quite a bit."

Republicans, however, took a dim view of Gore's report.

"This has all come trickling out over the past two years," said Rep. Rob Portman (R-Ohio), a co-chairman of the bipartisan National Commission on Restructuring the IRS. He said the section of the report aimed at curtailing abusive tax collection practices was first suggested by the commission in June 1997.

"But my larger point is that internal reforms are not going to solve the problem. We need legislation," said Portman, who advocates creation of an IRS oversight board composed of outsiders who would impose greater accountability on the agency.

Senate Finance Committee Chairman William V. Roth Jr. (R-Del.) said: "The problems at the IRS will not be corrected by reports alone. This study does not address the plight of taxpayers drowning under mounting interest and penalties. . . . Legislation needs to go beyond 'Reinventing Service at the IRS'—I intend to reinvent the IRS as a whole."

Gore agreed yesterday that legislation is needed and urged the Senate to quickly pass a bill so President Clinton can sign into law IRS reforms "for this tax season and for all seasons."

A general DATE program written by R. W. Bemer in 1980. It converts between the three major date forms and Julian Day in both directions.

Note that most of the program is for I/O and exhaustive explanations (with references).

Also note what a very small space is required for the actual calculations.

....

```
!date _ 1980-07-30 author:RWBemer, 602-942-1360
 scan:*date:"-" i="19",*i scan:*r:"-" j=*i k=*r count=0 case subs \
 out:"For today's date, do you want Ordinal, Fiscal, or Julian form?"
!again count=count+1 in:%Respond "ORD", "FIS", or "JUL"   %
 if count:lt:3 ergo !again call !cal_to_\*in\ \*svmd\ return
 out:"No action taken." nocase return

 !cal_to_cal
 !ord_to_ord
 !fis_to_fis
 !jul_to_jul out:"----No action---" n=1 call !explain2 return
 !cal_to_ord call !co call !mesg im=mc om=mo goto !reply
 !cal_to_fis call !cf call !mesg im=mc om=mf goto !reply
 !cal_to_jul call !cj call !mesg im=mc om=mj goto !reply
 !ord_to_cal call !oc call !mesg im=mo om=mc goto !reply
 !ord_to_fis call !of call !mesg im=mo om=mf goto !reply
 !ord_to_jul call !oj call !mesg im=mo om=mj goto !reply
 !fis_to_cal call !fc call !mesg im=mf om=mc goto !reply
 !fis_to_ord call !fo call !mesg im=mf om=mo goto !reply
 !fis_to_jul call !fj call !mesg im=mf om=mj goto !reply
 !jul_to_cal call !jc call !mesg im=mj om=mc goto !reply
 !jul_to_ord call !jo call !mesg im=mj om=mo goto !reply
 !jul_to_fis call !jf call !mesg im=mj om=mf goto !reply
 !reply out:" " out:\im\," is ",\om\
  \*svmd\ out:" " return

 !mesg subs \ mc=%i,"-",("0",j)['2,"-",("0",k)['2% mo=%"Day ",iday," of ",i%
  mf=%"Day ",fd," of FW ",fw," of FY ",fy% mj=%"Julian Day ",jd% return

 !julian1 call !jd1 out:"The first day of ",i," is Julian Day ",jd1 return
 !fiscal1 call !fd1 out:"The first day of ",i," falls on Fiscal Day ",fd1
  if fd1:gt:4 out:"But it's in Fiscal Year ",(i-1)
  return
 !fd1 call !jd1 fd1=jd1/7 fd1=*rmdr+1 return
 !calc_fc call !jd1 t=jd1-jd1/7*7 fc=t+7-(t+3)/7*7 return _ fc<7 means FW 0
 !jd1 jd1=1461*(i+4799)/4-31738-3*((i+4899)/100)/4 return

 !leap _ An easier way in TEX (· as substitute char) is:   l="if *rmdr:eq:0 leap"
  call !lp _ leap=0 q=i/4 ·i·=1 q=q/25 ·i·=0 q=q/4 ·i·=1 q=q/10 ·i·=0
  l=" isn't" l=i'[(3*leap) out:i,l,% a leap year. "leap"=%,leap return
 !lp leap=1-(i-i/4*4+3)/4+(i-i/100*100+99)/100-(i-i/400*400+399)/400
  t=i/4000 if *rmdr:eq:0 leap=0
  return

 !caldiff form="MM-DD" r="c" li=8 call !ask return
 !orddiff form="III"   r="o" li=7 call !ask return
 !fisdiff form="WW-D"  r="f" li=7 call !ask return

 !ask out:"Separators may be omitted."
 !d1 in:"First date (YYYY-",form,")?  " date1=*in if *lin:lt:li goto !d1
 !d2 in:"Second date?  " subs \ if *lin:lt:li goto !d2
  date2=*in call !\r\d out:"Difference is ",diff," days " \*svmd\ return

 !cd do=%split:arg:4 i=*i splitr:*r:2 k=*r j=(*i>'*n)'<*n call !cj%
  call !diff return
 !od do=%i=arg'14 iday=arg['3 call !oj%
  call !diff return
 !fd do=%split:arg:4 fy=*i splitr:*r:1 fd=*r fw=(*i>'*n)'<*n call !fj%
  call !diff return
```

```
!diff arg=date1 subs \
 \do\ firstj=jd arg=date2 \do\ diff=firstj-jd
 \*svmd\ return

!co call !lp iday=3055*(j+2)/100-(j+10)/13*2-91+leap*(j+10)/13+k return
!cf call !co call !of return
!cj jd=k-32075+1461*(i+4800+(j-14)/12)/4
 jd=jd+367*(j-2-(j-14)/12*12)/12-3*((i+4900+(j-14)/12)/100)/4 return
!oc call !lp id=iday+((305+iday-leap)/365)*(2-leap)
 j=((id+91)*100)/3055-2 k=id+30-(j*3056)/100 return
!of call !calc_fc fw=(iday+fc-1)/7 fd=*rmdr+1 fy=i
 if fw:eq:53 if (fc+leap):lt:10 fy=i+1 fw=1
 if fw:eq:0 i=i-1 fy=i call !lp i=i+1 fw=53-(fc+1-leap)/6
 return
!oj call !jd1 jd=jd1+iday-1 return
!fc call !fo call !oc return
!fo i=fy call !calc_fc iday=7*fw+fd-fc call !lp
 if iday:gt:(365+leap) i=i+1 iday=iday-365-leap
 if iday:lt:1 i=i-1 call !lp iday=365+leap+iday
 return
!fj call !fo call !oj return
!jc l=jd+68569 n=4*l/146097 l=l-(146097*n+3)/4 i=4000*(l+1)/1461001
 l=l-1461*i/4+31 j=80*l/2447 k=l-2447*j/80 l=j/11 j=j+2-12*l
 i=100*(n-49)+i+l return
!jo call !jc call !co return
!jf call !jc call !cf return

!explain_cal_to_ord out:"CAL_to_ORD" call !setup
 \calin\ \oo\,viday \ov\,vleap
 out:"Ref:     CACM 1972-10, p. 918, JDRobertson" \*svmd\ out:" " return
!explain_cal_to_fis out:"CAL_to_FIS" call !setup
 \calin\ \fisout\
 \ov\,viday \ov\,vfc \ov\,vleap
 \*svmd\ out:" " return
!explain_cal_to_jul out:"CAL_to_JUL" call !setup
 \calin\ \oo\,vjd
 out:"Ref:     CACM 1968-10, p.657, HFFliegel, TCVanFlandern"
 \*svmd\ out:" " return
!explain_ord_to_cal out:"ORD_to_CAL" call !setup
 \ordin\
 \oo\,vj \ov\,vk \ov\,vleap \*svmd\
 out:"Ref:     CACM 1970-10, p. 621, Stone (modified)" out:" " return
!explain_ord_to_fis out:"ORD_to_FIS" call !setup
 \ordin\ \fisout\
 \ov\,vfc \*svmd\ out:" " return
!explain_ord_to_jul out:"ORD_to_JUL" call !setup
 \ordin\ \oo\,vjd \*svmd\ out:" " return
!explain_fis_to_cal out:"FIS_to_CAL" call !setup
 \fisin\ \calout\
 \ov\,vleap \ov\,vfc \ov\,viday \*svmd\ out:" " return
!explain_fis_to_ord out:"FIS_to_ORD" call !setup
 \fisin\ \ordout\
 \ov\,vleap \ov\,vfc \*svmd\ out:" " return
!explain_fis_to_jul out:"FIS_to_JUL" call !setup
 \fisin\ \ordout\
 \ov\,vjd \ov\,vleap \*svmd\ out:" " return
!explain_jul_to_cal out:"JUL_to_CAL" call !setup
 \oi\,vjd \calout\
 out:"Ref:     CACM 1968-10, p.657, HFFliegel, TCVanFlandern"
```

```
 \*svmd\ out:" " return
!explain_jul_to_ord out:"JUL_to_ORD" call !setup
 \oi\,vjd \ordout\ \*svmd\ out:" " return
!explain_jul_to_fis out:"JUL_to_FIS" call !setup
 \oi\,vjd \fisout\ \*svmd\ out:" " return
!explain_caldiff out:"CALDIFF" call !setup
 \oi\,vc1 \ov\,vc2 \oo\,vdiff \*svmd\ out:" " return
!explain_orddiff out:"ORDDIFF" call !setup
 \oi\,vo1 \ov\,vo2 \oo\,vdiff \*svmd\ out:" " return
!explain_fisdiff out:"FISDIFF" call !setup
 \oi\,vf1 \ov\,vf2 \oo\,vdiff \*svmd\ out:" " return
!explain_julian1 out:"JULIAN1" call !setup
 \oi\,vi \oo\,"jd1  ",vj1," of Jan 01)" \*svmd\ out:" " return
!explain_fiscal1 out:"FISCAL1" call !setup
 \oi\,vi \oo\,"fd1  (D -- Fiscal Day for Jan 01)" \*svmd\ out:" " return
!explain_leap out:"LEAP" call !setup
 \oi\,vi \oo\,vleap \*svmd\ out:" " return

!setup subs \ ergo !go_on if vleap:eqs:vleap return
!go_on out:" " oi="out:%Inputs:  %" oo="out:%Outputs: %"
 vfc="fc   (fiscal constant -- year offset)" ov="out:%          %"
 vfd="fd   (D  -- Fiscal Day)" vfw="fw   (WW -- Fiscal Week)"
 vi="i    (YYYY -- year)" vj="j    (M or MM -- calendar month)"
 vk="k    (D or DD -- calendar day)" vfy="fy   (FFFF -- Fiscal Year)"
 viday="iday (XXX  -- ordinal day of the year)"
 vj1="(XXXXXXX -- Julian Day" vjd="jd   ",vj1,")"
 vleap="leap (1 if leap year, 0 if not)." vdiff="diff  (in days)"
 vc1=" (YYYY-MM-DD or YYYYMMDD)" vc2="date2",vc1  vc1="date1",vc1
 vo1=" (YYYY-III or YYYYIII)"    vo2="date2",vo1  vo1="date1",vo1
 vf1=" (FFFF-WW-D or FFFFWWD)"   vf2="date2",vf1  vf1="date1",vf1
 calin=",vi \ov\,vj \ov\,vk" calout=oo,calin calin=oi,calin
 ordin=",vi \ov\,viday" ordout=oo,ordin ordin=oi,ordin
 fisin=",vfy \ov\,vfw \ov\,vfd" fisout=oo,fisin fisin=oi,fisin return

!explain out:" " out:"DATE is a collection of calendar algorithms"
 out:"that use TEX or FORTRAN integer arithmetic" n=0
 out:"but little logic to calculate." out:" " g="       "
!explain2 out:"TEXLIB/U/DATE converts in every way between"
 out:"four forms of the date -- Calendar (CAL or C),"
 out:"Ordinal (ORD or O), Fiscal (FIS or F), or"
 out:"Julian (JUL or J), but not to the same." out:" " if n:ne:0 return
 out:"These are the entry points; the 2-letter form"
 out:"is for noninteractive subroutines:" out:" "
 out:"   CAL_to_ORD    CO",g," FIS_to_CAL    FC"
 out:"   CAL_to_FIS    CF",g," FIS_to_ORD    FO"
 out:"   CAL_to_JUL    CJ",g," FIS_to_JUL    FJ"
 out:"   ORD_to_CAL    OC",g," JUL_to_CAL    JC"
 out:"   ORD_to_FIS    OF",g," JUL_to_ORD    JO"
 out:"   ORD_to_JUL    OJ",g," JUL_to_FIS    JF" out:" "
 out:"   CALDIFF",g,"CD",g," JULIAN1",g,"JD1"
 out:"   ORDDIFF",g,"OD",g," FISCAL1",g,"FD1"
 out:"   FISDIFF",g,"FD",g," LEAP  ",g,"LP" out:g,g,"VC VO VF"
 out:" " out:"Explanations and input-output specifications are"
 out:"found by calling entry !EXPLAIN_(long form). E.g.,"
 out:" " out:"  CALL TEXLIB/U/DATE!EXPLAIN_CAL_to_JUL   (or)"
 out:"  CALL TEXLIB/U/DATE!EXPLAIN_ALL" out:" "
 out:"...DIFF (or .D) give the number of days between two"
 out:"dates given in calendar, ordinal, or fiscal form."
 out:*lf,"Input to this program may be validated prior to"
```

```
 out:"actual call by calling at VC, VO, or VF -- for"
 out:"Calendar, Ordinal, or Fiscal form respectively."
 out:"A variable VALID is returned as either T or F." out:" "
!end_explain out:" " return

!explain_all subs \ call !explain e="call !explain_"
 \e\cal_to_ord \e\cal_to_fis \e\cal_to_jul \e\ord_to_cal \e\ord_to_fis
 \e\ord_to_jul \e\fis_to_cal \e\fis_to_ord \e\fis_to_jul \e\jul_to_cal
 \e\jul_to_ord \e\jul_to_fis \e\caldiff \e\orddiff \e\fisdiff
 \e\julian1 \e\fiscal1 \e\leap \*svmd\ return

!vc valid="f" ergo !bad_input z=i+j+k ti=i tj=j tk=k
 call !co call !oc if j:le:12 if j:eq:tj if k:eq:tk valid="t"
 return
!vo valid="f" ergo !bad_input z=i+iday tiday=iday ti=i
 call !lp if iday:le:(365+leap) if iday:ge:1 valid="t"
 return
!vf valid="f" ergo !bad_input z=fy+fw+fd tfy=fy tfw=fw tfd=fd
 call !fj call !jf if fy:eq:tfy if fw:eq:tfw if fd:eq:tfd valid="t"
!bad_input return

!test in:"Date? (YYYYMMDD)  " t=*in split:*in:4 i=*l
 split:*r:2 j=*l k=*r jd="?" iday=jd fy=jd fw=jd fd=jd

!testloop i=i+1 date=i,("0",j)['2,("0",k)['2
 fd=fy,("0",fw)['2,fd out:t,"   ",iday,"   ",jd,"   ",fd
 call !co call !oj call !jf call !fc
 t=i,("0",j)['2,("0",k)['2 if t:nes:date out:"..",t
 call !cj call !jf call !fo call !oc
 t=i,("0",j)['2,("0",k)['2 if t:nes:date out:"..",t
 call !cf call !fo call !oj call !jc
 t=i,("0",j)['2,("0",k)['2 if t:nes:date out:"..",t
 call !co call !of call !fj call !jo call !oc
 t=i,("0",j)['2,("0",k)['2 if t:nes:date out:"..",t
 goto !testloop
```

```
-call texlib/u/date!explain_all

DATE is a collection of calendar algorithms
that use TEX or FORTRAN integer arithmetic
but little logic to calculate.

TEXLIB/U/DATE converts in every way between
four forms of the date -- Calendar (CAL or C),
Ordinal (ORD or O), Fiscal (FIS or F), or
Julian (JUL or J), but not to the same.

These are the entry points; the 2-letter form
is for noninteractive subroutines:

   CAL_to_ORD     CO        FIS_to_CAL     FC
   CAL_to_FIS     CF        FIS_to_ORD     FO
   CAL_to_JUL     CJ        FIS_to_JUL     FJ
   ORD_to_CAL     OC        JUL_to_CAL     JC
   ORD_to_FIS     OF        JUL_to_ORD     JO
   ORD_to_JUL     OJ        JUL_to_FIS     JF

   CALDIFF        CD                       VC
   ORDDIFF        OD                       VO
   FISDIFF        FD                       VF

   JULIAN1        JD1
                  FD1
   LEAP           LP

Explanations and input-output specifications are
found by calling entry !EXPLAIN_(long form). E.g.,

  CALL TEXLIB/U/DATE!EXPLAIN_CAL_to_JUL   (or)
  CALL TEXLIB/U/DATE!EXPLAIN_ALL

...DIFF (or .D) give the number of days between two
dates given in calendar, ordinal, or fiscal form.
Number of working days is in process.

Input to this program may be validated prior to
actual call by calling at VC, VO, or VF -- for
Calendar, Ordinal, or Fiscal form respectively.
A variable VALID is returned as either T or F.
```

```
CAL_to_ORD

Inputs:  i     (YYYY -- year)
         j     (M or MM -- calendar month)
         k     (D or DD -- calendar day)
Outputs: iday  (XXX  -- ordinal day of the year)
         leap  (1 if leap year, 0 if not).
Ref:     CACM 1972-10, p. 918, JDRobertson

CAL_to_FIS
Inputs:  i     (YYYY -- year)
         j     (M or MM -- calendar month)
         k     (D or DD -- calendar day)
Outputs: fy    (FFFF -- Fiscal Year)
         fw    (WW -- Fiscal Week)
         fd    (D  -- Fiscal Day)
         iday  (XXX  -- ordinal day of the year)
         fc    (fiscal constant -- year offset)
         leap  (1 if leap year, 0 if not).

CAL_to_JUL
Inputs:  i     (YYYY -- year)
         j     (M or MM -- calendar month)
         k     (D or DD -- calendar day)
Outputs: jd    (XXXXXXX -- Julian Day)
Ref:     CACM 1968-10, p.657, HFFliegel, TCVanFlandern

ORD_to_CAL
Inputs:  i     (YYYY -- year)
         iday  (XXX  -- ordinal day of the year)
Outputs: j     (M or MM -- calendar month)
         k     (D or DD -- calendar day)
         leap  (1 if leap year, 0 if not).
Ref:     CACM 1970-10, p. 621, Stone (modified)

ORD_to_FIS
Inputs:  i     (YYYY -- year)
         iday  (XXX  -- ordinal day of the year)
Outputs: fy    (FFFF -- Fiscal Year)
         fw    (WW -- Fiscal Week)
         fd    (D  -- Fiscal Day)
         fc    (fiscal constant -- year offset)

ORD_to_JUL
Inputs:  i     (YYYY -- year)
         iday  (XXX  -- ordinal day of the year)
Outputs: jd    (XXXXXXX -- Julian Day)

FIS_to_CAL
Inputs:  fy    (FFFF -- Fiscal Year)
         fw    (WW -- Fiscal Week)
         fd    (D  -- Fiscal Day)
Outputs: i     (YYYY -- year)
         j     (M or MM -- calendar month)
         k     (D or DD -- calendar day)
         leap  (1 if leap year, 0 if not).
         fc    (fiscal constant -- year offset)
         iday  (XXX  -- ordinal day of the year)
```

```
FIS_to_ORD
Inputs:  fy    (FFFF -- Fiscal Year)
         fw    (WW -- Fiscal Week)
         fd    (D  -- Fiscal Day)
Outputs: i     (YYYY -- year)
         iday  (XXX  -- ordinal day of the year)
         leap  (1 if leap year, 0 if not).
         fc    (fiscal constant -- year offset)

FIS_to_JUL
Inputs:  fy    (FFFF -- Fiscal Year)
         fw    (WW -- Fiscal Week)
         fd    (D  -- Fiscal Day)
Outputs: i     (YYYY -- year)
         iday  (XXX  -- ordinal day of the year)
         jd    (XXXXXXX -- Julian Day)
         leap  (1 if leap year, 0 if not).

JUL_to_CAL
Inputs:  jd    (XXXXXXX -- Julian Day)
Outputs: i     (YYYY -- year)
         j     (M or MM -- calendar month)
         k     (D or DD -- calendar day)
Ref:     CACM 1968-10, p.657, HFFliegel, TCVanFlandern

JUL_to_ORD
Inputs:  jd    (XXXXXXX -- Julian Day)
Outputs: i     (YYYY -- year)
         iday  (XXX  -- ordinal day of the year)

JUL_to_FIS
Inputs:  jd    (XXXXXXX -- Julian Day)
Outputs: fy    (FFFF -- Fiscal Year)
         fw    (WW -- Fiscal Week)
         fd    (D  -- Fiscal Day)

CALDIFF
Inputs:  date1 (YYYY-MM-DD or YYYYMMDD)
         date2 (YYYY-MM-DD or YYYYMMDD)
Outputs: diff  (in days)

ORDDIFF
Inputs:  date1 (YYYY-III or YYYYIII)
         date2 (YYYY-III or YYYYIII)
Outputs: diff  (in days)

FISDIFF
Inputs:  date1 (FFFF-WW-D or FFFFWWD)
         date2 (FFFF-WW-D or FFFFWWD)
Outputs: diff  (in days)

JULIAN1
Inputs:  i     (YYYY -- year)
Outputs: jd1   (XXXXXXX -- Julian Day of Jan 01)

FD1
Inputs:  i     (YYYY -- year)
Outputs: fd1   (D -- Fiscal Day for Jan 01)

LEAP
Inputs:  i     (YYYY -- year)
Outputs: leap  (1 if leap year, 0 if not).
```

BMR SOFTWARE

1997 Dec 05

Senator Robert Bennett:
Representative Stephen Horn:

This coming Wednesday there is a "Government Y2K" conference in Washington. The reports may say what you already know. But my keynote talk is new, and may be of substantial interest.

It is hereby faxed to you because you may wish to be aware of its content, because the recommendations coincide with your own, and because you may wish to have this advance notice (no others do) to see how it may fit with your own strategies.

The welcome shows that the Government is so far only addressing the programming problem, not the problem for people.

The technical part largely shows the ineffectiveness and dangers of current strategy. Skip, if you wish. But don't miss the beginning of the keynote, and the final page.

R. W. Bemer

. . .

11300 N. Central Expwy., Suite 404 • Dallas, Texas 75243 • (214) 739-3200 • Fax (214) 739-0848

ALLERTON CUSHMAN
R.R. #1, Box 67, Seacrest,
Nanoose Bay, B.C.
Canada V0R 2R0

V8R 4B0 971110 23:44

Air Mail Par avion

LUNG ASSOC. PULMONAIRE

Mr. Bob Bemer
MBR Software #400
5930 LBJ Freeway,
DALLAS, TX.
75240

Personal

240/6315

Bob Bemer

From: Bruce Webster <g8ubew@fanniemae.com>
To: FMY2K@fanniemae.com; WDCY2K@fanniemae.com
Cc: OSGALL@fanniemae.com
Subject: Yourdon on Washington/Y2K (resend)
Date: Wednesday, December 24, 1997 8:13 AM

My mail application has a sporatic bug that causes it to uuencode messages that have been forwarded into the mail I send out. My apologies. I hope this gets through OK. ..bruce..

Bruce F. Webster, CTO, Object Systems Group
Member, Fannie Mae Year 2000 Team
Chair Pro Tem, Washington DC Year 2000 Group
email: bruce_webster@fanniemae.com
voice: 202.752.3979
pager: 800.516.3358
web: http://www.bfwa.com/bwebster/y2k

Welcome to The Y2000 E-Mail Advisor, a weekly electronic briefing from Ed Yourdon, Director of the Cutter Consortium's Y2000 Advisory Service.

SAYONARA, WASHINGTON

Nobody seems willing or able to say it in simple language, so let me be the one: the federal government is not going to finish its Y2000 project. No maybes, no ifs, ands, or buts. No qualifiers, no wishy-washy statements like "unless more money is spent" or "unless things improve." We're not going to avert the problem by appointing a Y2000 Czar or creating a National Y2000 Commission. Let me say it again, in plain English: The United States federal government will not finish its Y2000 project.

If there was any previous uncertainty about this, the December 11 update from Congressman Stephen Horn, who chairs the House Government Management, Information and Technology Subcommittee, leaves no doubt about the outcome. While 10 of the 24 major federal agencies are currently claiming that they'll be done in time, 14 will not. Horn's report says that based on current rates of progress (which almost always turn out to be hysterically optimistic in the early stages of a software project!), these 14 agencies can expect to finish Y2000 remediations of their "mission-critical" functions by the following dates:

2019: Energy Department

2019: Labor Department

2012: Defense Department

2010: Transportation Department

2010: Office of Personnel Management

2005: Agriculture Department

2004: Treasury Department

2002: General Services Administration

2001: Health and Human Services Department

2001: Justice Department

Mid-2000: Education Department

Mid-2000: Agency for International Development

Mid-2000: Federal Emergency Management Agency

Early 2000: National Air and Space Administration

And even this doesn't convey the extent of the problem. Horn's estimates account only for mission-critical systems being REPAIRED. There are many other mission-critical systems that are being REPLACED, either with new in-house systems development projects, or by purchasing a commercial package. I personally know of several "replacement" projects that are guaranteed to miss the 1/1/2000 deadline, but they don't show up in Horn's report at all -- indeed, the problems and delays associated with these replacement projects aren't showing up in ANY reports, for all the usual reasons of politics and bureaucracy (e.g., external contractors, who will deny they're behind schedule because it might cause their contract to be canceled). Also, the much larger number of non-mission-critical systems, the collective impacts of which are almost certain to have staggering consequences, are not in the report. Nor are the embedded systems -- e.g., the elevators, the PBX telephone systems, the security systems, the HVAC systems, etc.

Nor is there any mention of the "supply chain" of vendors and customers, or what the IRS euphemistically refers to as its "trading partners." It may seem, at times, that the government operates in a world unto itself, but the reality is that it depends very much on the private sector. If the computer hardware and software vendors fail in their efforts to deliver Y2000-compliant products, how can the federal government expect to succeed in its Y2000 project? If there are disruptions in utilities, banking/finance, or telecommunications, the impact will be felt in Washington, DC, just as much as in the "real world."

A few years ago, a budget impasse between Bill Clinton and the Congress effectively halted all of the non-mission-critical functions of government for a few weeks -- and it was intriguing to see that, by and large, it didn't cause much of an impact throughout the country. Similarly, if it turns out that NASA shuts down for a few months in 2000 because of Y2000 problems, it might not concern us; a few space launches might be canceled or deferred, but society would continue to function. But if Health and Human Services can't get their food-stamp and Medicare/Medicaid systems functioning properly until 2001, there will be riots in the streets. If it takes 19 years beyond the Y2000 deadline to get the Labor Department and the Energy Department to function properly, then why do we even need to have a Labor Department

or Energy Department? Can anyone seriously expect that the nation will hold its breath for 19 years while the programmers continue the endless task of fixing and testing the code?

How Washington expects to continue functioning after 1/1/2000 is a mystery to me. How American society expects to continue operating in a "business as usual" fashion, when half of the federal government agencies stop functioning, is a deeper mystery -- and one for which we must all begin planning. The contingency plans that we develop for ourselves and our families are beyond the scope of this weekly column, though it's precisely the subject that my daughter and I address in our "Time Bomb 2000" book. But it's also an area that requires contingency planning in the business sector: the CEO and the Y2000 project team must now take into account the virtual certainty that half of the US federal government agencies will not be functioning after New Year's Eve in 1999. By a similar argument, it's likely that the same fate awaits state and local government agencies.

The superficial reaction, on the part of many business executives, is to cheer: After all, government is regarded as a nuisance, an obstacle, and a drain on the resources of most companies. But whether we like to admit it or not, individuals and companies DO rely on government agencies, to a greater or lesser extent. At the local level, for example, we take for granted the availability of police, fire departments, and public transportation; if those services are disrupted for a month or a year, what impact will it have on our businesses? At the federal level, many companies depend on grants, subsidies, and contracts; how long will the Beltway Bandits and the aerospace industry last if there are no payments by the 14 non-compliant agencies for the first few years of the new decade?

All of this is so mind-boggling that it falls into the category of "thinking about the unthinkable." I don't like to think about it any more than anyone else, but it's unavoidable at this point. Realistically, we can no longer talk about what might happen IF Washington fails to fix its Y2000 problems. Realistically, we have to start talking about what will happen WHEN the Y2000 problem brings the government to its knees.

Realistically, we're only two years and a week away from the day when both citizens and business organizations will have to say, "Sayonara, Washington."

Ciao!
Ed

++++++++++++++++++++++++++++++++++

If you'd like to comment on today's Y2000 E-Mail Advisor, send e-mail to yourdon@cutter.com, or send a letter by fax to +1 781 648 8707 or by mail to The Y2000 E-Mail Advisor, Cutter Consortium, 37 Broadway, Arlington, MA 02174-5552 USA
++++++++++++++++++++++++++++++++++

Get a FREE copy of Ed Yourdon's white paper, *Y2000: Fear and Loathing*, when you complete the Consortium's Y2000 Compliance Survey online at <http://www.cutter.com/consortium/y2ksurvy.htm
++++++++++++++++++++++++++++++++++

From: Jeri Clausing <jeri@nytimes.com>
To: Bob Bemer <bbemer@bmrsoftware.com>
Date: Tuesday, March 10, 1998 11:35 AM
Subject: Re: Your Y2K article

thanks for pointing out the misspelling. somewhere along the line yesterday that got transposed.

i will look at your site before next week's hearing. .NEG DM, Pres NP Radio Guy

jeri

At 11:18 AM 3/10/98 -0600, you wrote:
> www.bmrsoftware.com "" Click on the "". That's the lie
>they put out two years ago. And if you really want to know what needs
>to be done in the way of a tsar (or does the NYT favor czar?) click on my
>Government keynote talk captioned there. Bob Bemer
>bbemer@bmrsoftware.com
>http://www.bmrsoftware.com

From: Bob Bemer <bbemer@bmrsoftware.com>
To: jeri@nytimes.com <jeri@nytimes.com>
Date: Tuesday, March 10, 1998 11:18 AM
Subject: Your Y2K article

Apart from the point that I thought the man's name was Koskinen (from Finnish), please check our WebVenue www.bmrsoftware.com to see my opinion of the "tsar" position. Click on the balloon "Dear Sen. Bennett".

Where did you get that old $2.3 billion estimate. That's the lie they put out two years ago.

And if you really want to know what needs to be done in the way of a tsar (or does the NYT favor czar?) click on my Government keynote talk captioned there.

Bob Bemer
bbemer@bmrsoftware.com
http://www.bmrsoftware.com

TO WEB –
FOR RECORD

Washington Trip Report -- Bob Bemer

Yes, I went to Washington to present my concerns and suggestions to Senator Bennett (R. Utah) at the meeting of the DCA Y2K Users Group at Fannie Mae headquarters on March 18. Yes, I was checked in and given a preprinted name badge (stick-on type). Yes, I met Bruce Webster, the continuing Chair of this group, and a contract employee of Fannie Mae. I got a very small amount of my income taxes back by accepting two glasses of white wine and some calamari at a quite nice buffet adjacent to the "Great Hall".

Then I became disconcerted. Mr. Webster informed me that the Senator probably wouldn't wish to be bothered with my suggestions, which he (Webster) said he had read. Moreover, he wanted me to stop any further distribution of the handout containing my ideas, because the Government prohibited that on their premises (even for ideas you are trying to give them free of charge in this crisis).

Suddenly my self-confidence broke, It was obvious that all of these U.S. Government people knew that they were smarter and better than I was, even though they have not been able to fix the problem themselves. Even the cabdrivers I met echoed this conclusion. So I left before the Senator arrived, so as to not embarrass him. The next day I gave my placards and copies to the Washington Post writer who had covered the House hearings the previous day, where new Czar John Koskinen advised everyone not to panic -- it would all turn out OK just in time. This writer may throw away my hard work, to make date exchange rigorous and easy, in similar disdain. I just don't know, now that I've lost my self-assurance.

But won't you judge my proposal yourself? It follows!

UNIVERSAL DATE VALUE INTERCHANGE

R. W. Bemer, BMR Software (www.bmrsoftware.com)

Computers do only three things with date values. They

1) Take them in,
2) Do calculations and other manipulations based upon them.
3) Put them out, to display, control, or another computer.

(2) is the tough part, where most of the Y2K problems lie, due to

- Programs that operate upon them in myriad ways, often so poorly documented as to be ununderstandable, and
- Again, their insufficient form (missing century).

(1) and 3) present more problems. The insufficiency, yes, but also the format/layout. Is a date for import (1) or export (3)

- In YYMMDD or MMDDYY or DDMMYY form?
- In YYYYMMDD or MMDDYYYY or DDMMYYYY form?
- Encoded in decimal or binary numbers?
- Encoded in some symbolic form other than direct numbers?
- If decimal, 8 bits or 4 bits per digit of date?
- If discrete binary, 16 bits or 8 bits per date unit?
- If full binary, 32 bits or 16 bits per complete date?
- Relative to some other date by a number of days?
- Interspersed with delimiters and/or spaces?

But the big question is -- with all these possible and existing uses, how do we know which? More properly, how does a stranger know? We know, of course, because it's all implicit in the programs we are using.

WE CAN MOSTLY FIX (1) AND (3) FOR INTERCHANGE RIGHT NOW!

HERE IS HOW

Only one unfailing human method of demarking time exists. The earth rotates once each day. Has and will. Two more rotations make it two days later. Thus there is only one basic way to know the date. How many days is it from some starting date? Unsurprisingly such a method exists. It is called the Julian Day system. The Julian day for the first day of Year 1 A.D. had the value:

0001 = 1 721 475

For day 1 of the following centuries the Julian Day was/will be:

400 = 1 867 157	1700 = 2 341 973	2100 = 2 488 070
800 = 2 013 254	1800 = 2 378 497	2200 = 2 524 594
1200 = 2 159 351	1900 = 2 415 021	2300 = 2 561 118
1600 = 2 305 448	2000 = 2 451 545	2400 = 2 597 642

The leading digit doesn't change for 27 centuries. 12 centuries with a leading "2" have passed, and we have 15 more to go before the "2" rolls over to "3". So we don't need the leading "2". Imply it, and you ALMOST get what is called the Smithsonian Day (which is that value + 1). Call this value the Xchange Day (X Day). It has some lovely properties:

- Scrap all of your leap year code. The 100- and 400-year leap year exceptions are built into the conversion formulas.
- To get the day of the week, add 2 to the X Day value, divide by 7, and add 1 to the remainder (Monday=1).
- One can't mistake the first two digits as month or day values, etc.
- It needs the same space -- 6 zoned-, or 4 signed packed decimal-, or 3 unsigned packed decimal-bytes -- to accommodate 27 centuries as do separate 2-digits for year and month and day do for just one century. Don't change your record formats; fill them differently.
- This would accommodate the current minimum representation, due to Equifax, of 3 bytes, one each for discrete binary day, month, year. Full binary, which gets 16777216 in 3 bytes, can get only 65536 in 2 bytes, for a span of at most 179 years, which is not good enough.
- Vertex 2000 TM has the same minimums, and we need that internally for now, but V2K dates are just as easily converted to X Days.

Let's make the X Day the gold standard, the lingua franca, the common denominator, the canonical form -- our salvation in the world of data interchange. Use it like Euro currency, one unit of which should be equivalent to so many lira, or dollars, or pounds, or francs, or rubles. But the Euro is artificially derived; days come from the real world.

So no matter what you use at home, in the world market you can use the X Day exclusively to represent the date.

What does that mean to us in the present crisis? We know it won't fix all programs an entity uses to run its enterprise, but it will surely work as the default medium of exchange. And exchange has two properties:

1) Businesses can most likely put an exact finger on the character of dates they input and output externally.
2) Everyone now realizes the greatest danger of the 2-digit year to be nonstandardized and unrecognizable interchange, which this cures.

Can the United States and the rest of the world profit NOW by using only the X Day for interchange?

- Will this solve the Year 2000 problem? Definitely not.
- Will it take the sting out of our coming collapse? Definitely yes.
- Can it be done in time? With the right authority, a good chance.

What must be done? First we must get agreement to standardize this. If we wait for the GSA or business groups to agree, forget it! But suppose

the Congress passes a law like this?

> In date data interchange under private agreement between exporter and importer, both may represent the date in any way they agree to.
>
> Absent such agreement, electronic interchange of year values must be done only in X Day form.

Converting X Day to all other forms, simple and compound, is easy. The formulas are known and simple (but I have copied them in an Appendix). I had planned for my company to make available to all, as a free public service, source and object computer programs, in the common computer languages, for translation to and from X Day from other forms such as calendar, ordinal, and fiscal dates.

But others have a huge debt to us users for being at fault for Y2K. Let's ask Microsoft to provide these as their gift to ameliorating this crisis, in compensation for having done it wrong in the first place. Imagine sitting at your PC:

```
C:\>time
Current time is 2:17:31.29p
Enter new time:

C:\>date
Current date is Wed 03-18-1998
Enter new date:

C:\>xday
Current X Day is 450891

C:\>dayo                              (or maybe we choose this name)
Bad command or file name
```

Let's ask Sun to contribute free Java applets for these rules. Let's ask IBM and UNISYS and others to provide all such routines free for their computers. They should respond gladly and soon. Or is patriotism dead? Or is self-interest going to be totally destructive?

Such a package would be applied by the sender just before sending, and by the receiver to convert to the form they need. Instead of talking face-to-face, interpose two translating telephones. Then when and if the standards people ever get around to deriving what I suggested in my White Paper to the GSA, they can use any other date form for which an escape sequence is registered.

Would I wish that every computer in the world processed dates in X Day form? I certainly would. It's absolutely the simplest and best way. And one would hope that for the future, once the present 2000 crisis is passed, that they all would be so programmed. I think it as critical a standard to set as ASCII (the ISO Code).

(For questions, call Bob Bemer at 972-671-5000)

APPENDIX I -- X DAY CONVERSION FORMULAS (integer arithmetic!)

XD=X Day value	Y=4-digit year	yyyy	FY=fiscal year
JD=Julian Day value	M=2-digit month	mm	FW=fiscal week
JD1=JD for January 01	D=2-digit day	dd	FD=fiscal day
S,T=working	OD=ordinal day		FC=fiscal constant

My 1980 TEX program "DATE" adopted the following plan for conversions (that they may be further compacted to remove the "IF"s is obvious):

A)	CAL_to_JD	do (1)	
B)	CAL_to_JD1	do (2)	(for 1st day of CAL year)
C)	CAL_to_FIS	do (D) and (F)	
D)	CAL_to_ORD	do (4) and (5)	(CAL year known)

```
E)  ORD_to_CAL          do (4) and (6)                  (CAL year known)
F)  ORD_to_FIS          do (7) and (9)
G)  ORD_to_JD           do (2) and (JD=JD1+OD-1)

H)  FIS_to_ORD     Y=FY do (7) and (8)                    JD=XD+2000000
I)  FIS_to_CAL          do (H) and (E)                    XD=JD-2000000
J)  FIS_to_JD           do (H) and (G)

K)   JD_to_CAL          do (3)
L)   JD_to_FIS          do (K) and (C)
M)   JD_to_ORD          do (K) and (D)

1)    S=(M-14)/12 ... T=D-32075+1461*(Y+4800+S)/4
     JD=T+367*(M-2-S*12)/12-3*((Y+4900+S)/100)/4

2)  JD1=1461*(Y+4799)/4-31378-3*((Y+4899)/100)/4

3)    T=JD+68569
      S=4*T/146097
      T=T-(146097*S+3)/4
      Y=4000*(T+1)/1461001
      T=T-1461*Y/4+31
      M=80*T/2447
      D=T-2447*M/80
      T=M/11
      M=M+2-12*T
      Y=100*(S-49)+Y+T

4) LEAP=1-(Y-Y/4*4+3)/4+(Y-Y/100*100+99)/100-(Y-Y/400*400+399)/400
      T=Y/4000 if remainder:eq:0 LEAP=0                    (optional)

5)   OD=3055*(M+2)/100-(M+10)/13*2-91+LEAP*(M+10)/13+D

6)    T=OD+((305+OD-LEAP)/365)*(2-LEAP)
      M=((T+91)*100)/3055-2
      D=T+30-(M*3056)/100                          ----------------------
                                                  | Wow!  It would be so |
7)    do (2) ... T=JD1-JD1/7*7                    | easy to build a chip |
     FC=T+7-(T+3)/7*7                             | for these conversions |
                                                   ----------------------

8)   OD=7*FW+FD-FC ... do (4)
     if OD:gt:(365+LEAP) Y=Y+1 OD=OD-365-LEAP
     if OD:lt:1 Y=Y-1 do (4) OD=365+LEAP+OD

9)   FY=Y ... FW=(OD+FC-1)/7
     FD=remainder+1
     if FW:eq:53 if (FC+LEAP):lt:10 FY=Y+1 FW=1
     if FW:eq:0  Y=Y-1 FY=Y do (4) Y=Y+1 FW=53-(FC+1-LEAP)/6
```

APPENDIX II -- PROBABILITIES FOR XDAYS

A major Year 2000 problem is recognizing a date value by its name (in source code) or by its numeric value (in data). The first aspect is well-known to be very difficult. Laments are continual and loud.

Some programs exist to find 6-digit fields that conform to the rules that, in whatever combinatorial order, 1 of the 3 pairs has no value > 12, and another has no value > 31. Except, of course, if the field might be for an ordinal date (2 digits for year, and 3 digits for day of year). The "which order" is what makes all this more difficult. And of course the date values might be embedded (thus not permissible to change) within a field like an insurance policy number [1].

Now apply recognition rules to 6-digit fields where Xdays might exist. March 18 of 1998 is Xday 450891. That starting "45" won't roll over to "46" for another 9109 days -- about 25 years in the 27-year cycle.

Jan 01 of 1950, when very few computers were in use, was Xday 433283,

so Xday for almost ALL of today's computational needs will start with 43, 44, or 45. If we look at just the first digit of an Xday, a person would have had to be born before 1859 for it to be a "3". Xdays won't start with a "5" until 2132!

What other 6-digit data fields begin with only 3 of 100 combinations (43-45) and yet are unlikely to be the years 1945 or 2043? [2]

And note these welcome facts:

- Xday for 1998 Mar 18, in packed decimal form, is "04 50 89 1+". Four octets or bytes, 32 bits, just exactly one common 32-bit word.

- Think of how often we need the difference in days between two dates. Else why are ordinal dates so common (and often used so wrongly).

- Old 2-digit year values, mistakenly fed to the conversion formulas, give Xdays with negative values. Obvious enough? You've found one!

- We'd still use human forms for dates, but only for input and output from and to humans. All interchange between computers would be done in this standard Xday unit that they would all understand. Later, internal calculations and storage could use Xdays, too, in a gradual and manageable changeover.

Notes: 1. Whether people would begin to use Xdays there is moot.
2. Perhaps part numbers, but they can have letters, too.

BigiSoft Overview

AGENDA

- Introduction to BigiSoft Inc.
- Incorporation
- Distribution Strategy
- Management Team

BigiSoft, Inc.

Vertex 2000 Technical Review

AGENDA

• BigiSoft Overview	Roger Hughes
• Vertex 2000 Product Overview	Chuck Harvey
• Q&A Discussion	All

Introduction to BigiSoft, Inc.

- Tightly Focused on the Y2K Market
- Software Company Only with Unique Approach
- IBM MVS COBOL
- Vertex 2000 is not a "silver-bullet"
  - Interim solution
  - Source code independent
  - Short implementation cycle

Incorporated April 1998

- Acquired all assets of BMR Software - Vertex 2000
- Bob Bemer employed as Chief Scientist
- Close Relationship with SSA
  - Contract software development company
  - Irv Overman President and CEO
  - Employees share in BigiSoft stock

Distribution Strategy

- Use existing distribution channels
- Y2K Systems Integrators

Management Team

■ Ron Brittian	Chairman & CEO
■ Bob Bemer	Chief Scientist
■ Roger Hughes	CFO & COO
■ Chuck Harvey	Chief Technical Officer
■ Irv Overman	President of SSA
■ Hired	VP Sales
■ Hired	VP Marketing

BigiSoft, Inc.

Vertex 2000 Product Overview

AGENDA

- Introduction to Vertex 2000
- Pilot Project Implementation Process
- BETA site results to date
- Request for Pilot Project Opportunity

Introduction to Vertex 2000

- Patented Year Format - **Ver**tical **Ex**tension
- Packs 4 digits of information in 2 digits
- Called "Bigits" or "Bemer Digits" for Bob Bemer
- Benefits of this concept
  - No need to increase field or file lengths
  - No need to modify program source code
  - Reduces testing by reducing risk
  - V2K environment automatically recognizes Bigits

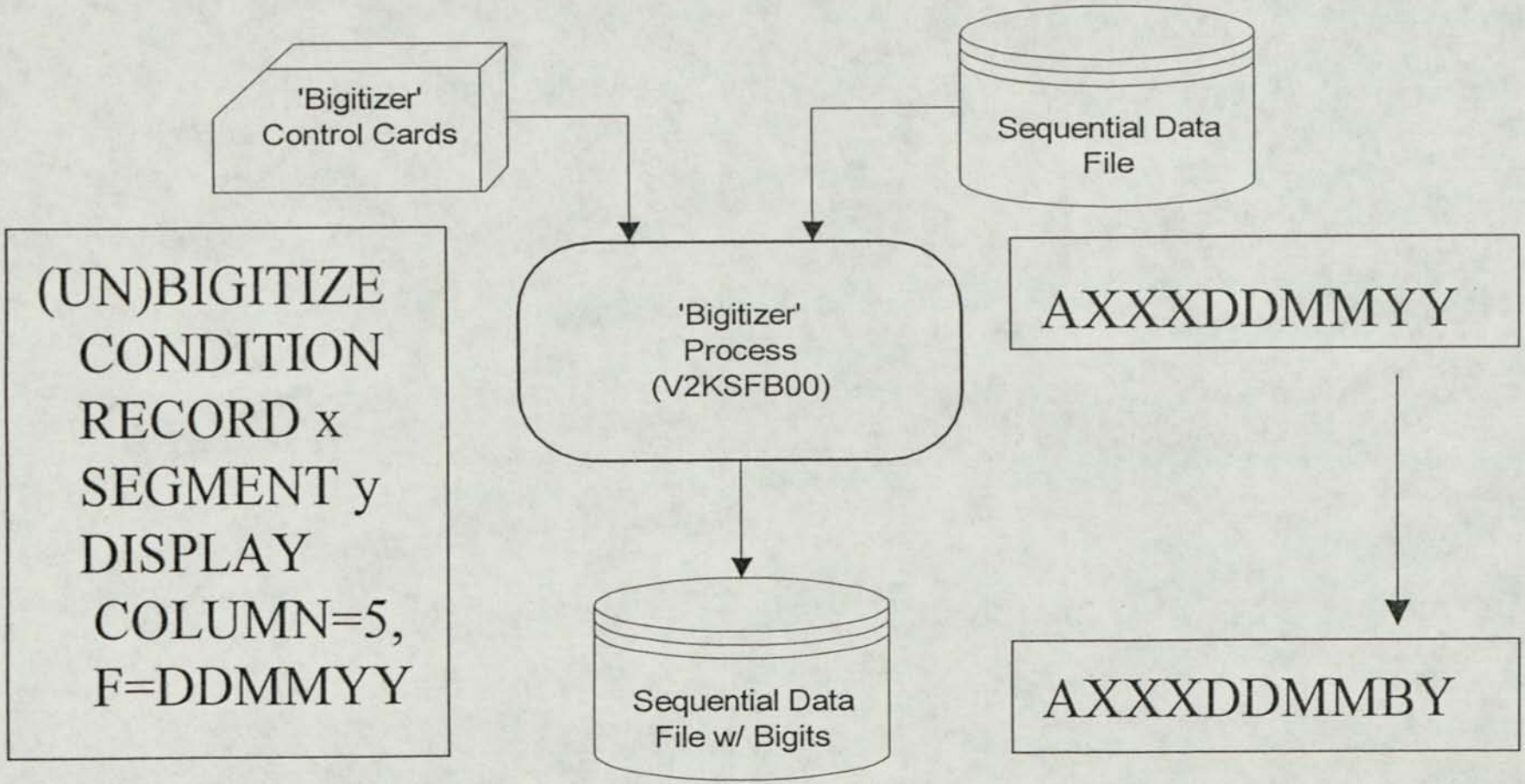
Where Vertex 2000 works

- IBM mainframes
- MVS operating systems
  - ESA
  - XA
- IBM COBOL compilers
  - COBOL II Release 4.0
  - COBOL/VS Release 2.4
  - COBOL for MVS (LE) Version 1 Release 1 Mod 0
- CICS - Release 1.7 and up

Product Status

	June	July	August
■ COBOL II	X		
■ COBOL/VS & LE			X
■ Data Bases			
◆ Flat Files	X		
◆ VSAM	X		
◆ Future data bases as needed to support market			

Bigitizing Data



Vertex 2000 Bigits

- A Bigit is a code that can be hidden inside the 2 digit year value using vertical extension

5 = 1600s

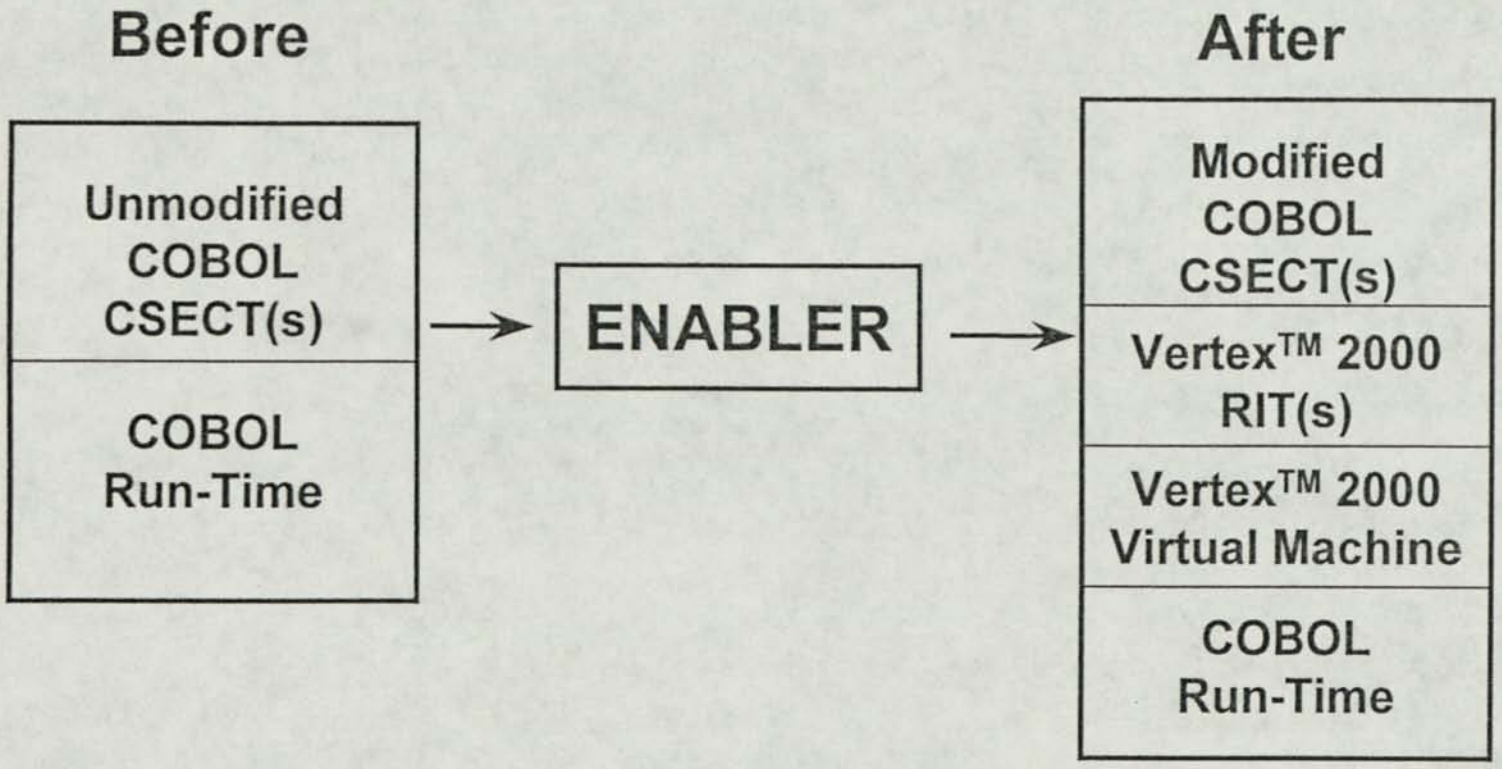
6 = 1700s

7 = 1800s

8 = 1900s

9 = 2000s

Enabler Actions



Traps are based on operands

Supported Date Formats

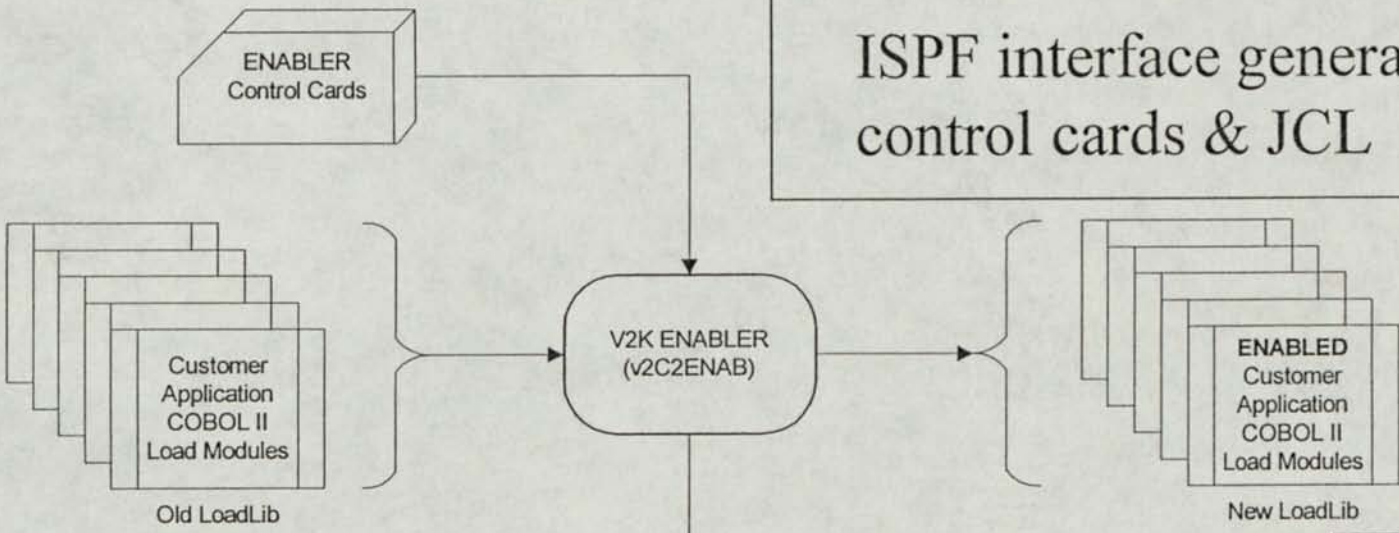
The Vertex 2000™ Enabler and Virtual Machine both support the following date formats:

Format	Zoned Decimal[1,2]	Packed Decimal	Binary[3]
YY	99	99 COMP-3	9(4) COMP
YYMM	9(4)	9(4) COMP-3	9(4) COMP
MMYY	9(4)	9(4) COMP-3	9(4) COMP
YYDDD	9(5)	9(5) COMP-3	
MMDDYY	9(6)	9(6) COMP-3	
YYMMDD	9(6)	9(6) COMP-3	
YYDDMM	9(6)	9(6) COMP-3	

Enabler traps date instructions

- Guilty Until Proven Innocent
  - All potential date-related instructions are trapped
  - Unorthodox formats need special attention
    - Application Programming Interface (API) provided
    - No different than current situation
    - Normally more than 90+% of date types covered
  - Internal application knowledge is not required
  - Specific date formats trapped
    - "Untrap" utility is provided to improve performance
    - High performance applications can be fine tuned

Enabling Programs



ISPF interface generates all control cards & JCL

Original modules are never harmed.

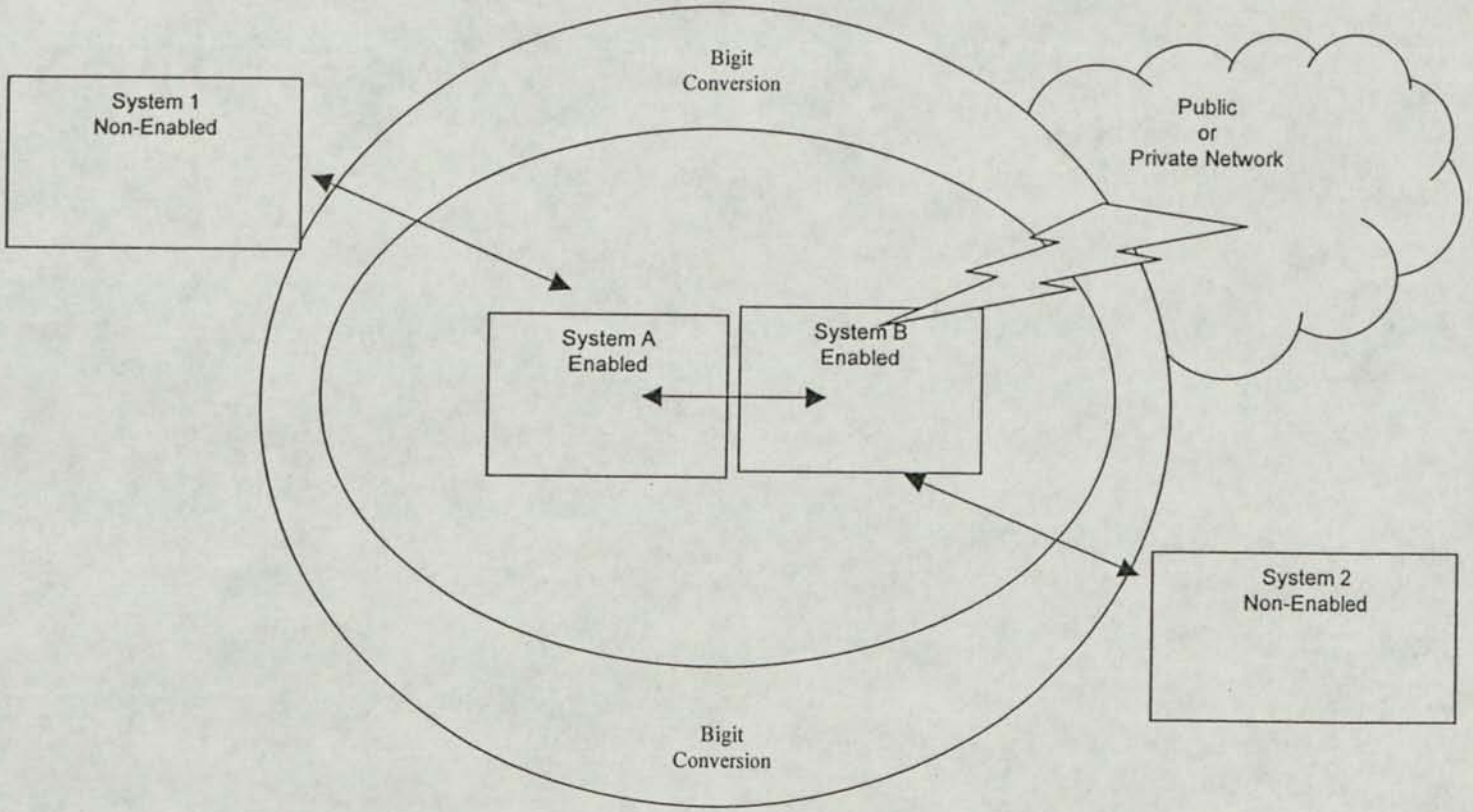
Audit Trail for all Enabler Actions

Vertex 2000 is a new environment

- Recognizes "bigits" when encountered
- Performs appropriate "bigit" arithmetic and comparisons as needed.
- Bypasses and coexists with correct non-bigit dates
- Program logic is not affected in any way
- Interfaces to the environment must be identified
  - 2 digit year entering must be "bigitized"
  - 2 digit year leaving must be "unbigitized"

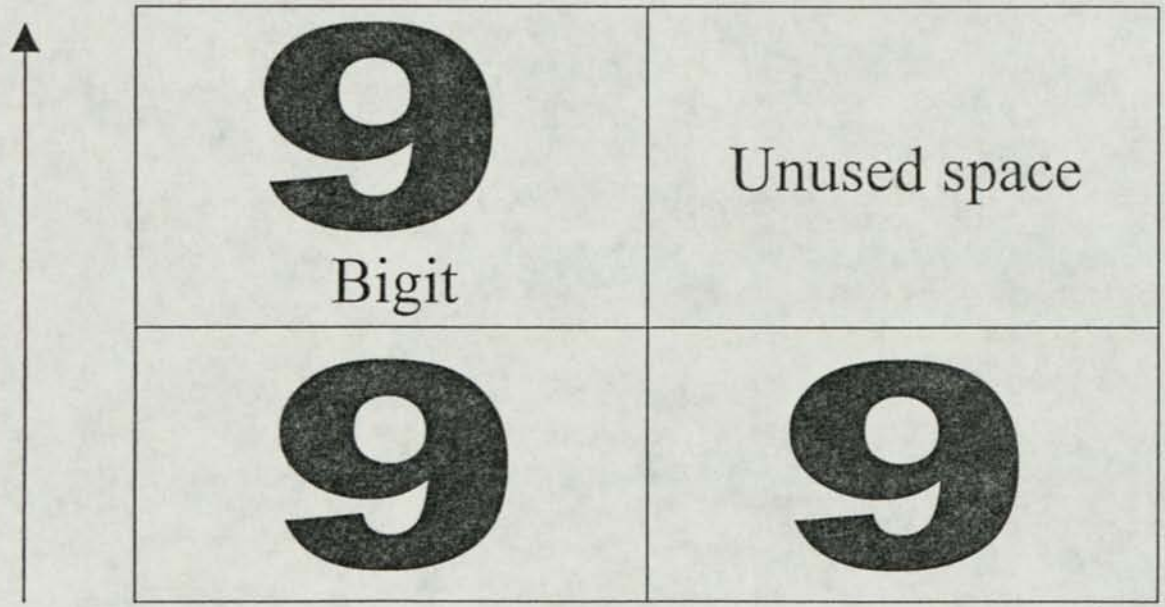
BigiSoft, Inc.

Vertex 2000 Environment

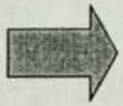
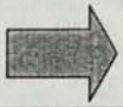
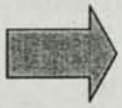
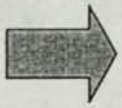


Bigit Example

- 2099 would be represented as:



Lower level examples

EBCDIC		Display	B-Packed	Value
X'F7F6'		X'67F6'	X'0766'	1776
X'F6F8'		X'76F8'	X'0687'	1868
X'F9F7'		X'89F7'	X'0978'	1997
X'F0F3'		X'90F3'	X'0039'	2003

Vertex 2000 Implementation

- What steps are needed to implement Vertex 2000
  - Product Installation
    - SMP/E or Standard IBM Utility Install
    - No system modifications or SVCs
  - System Remediation
    - ISPF-based Utility tool set provided for all phases
  - System Test
    - Automated direct comparison of baseline tests
  - Release, Maintain and Support

System Remediation*

Action	Who	Est. Time
■ Analysis	All	5 days
■ Create Test Baseline	End User	1 day
■ Enable Programs	End User	5 days
■ Bigitize Data	End User	5 days
■ Test System	End User	5 days
Total		21 days

* Nominal System e.g. 50 - 100 programs

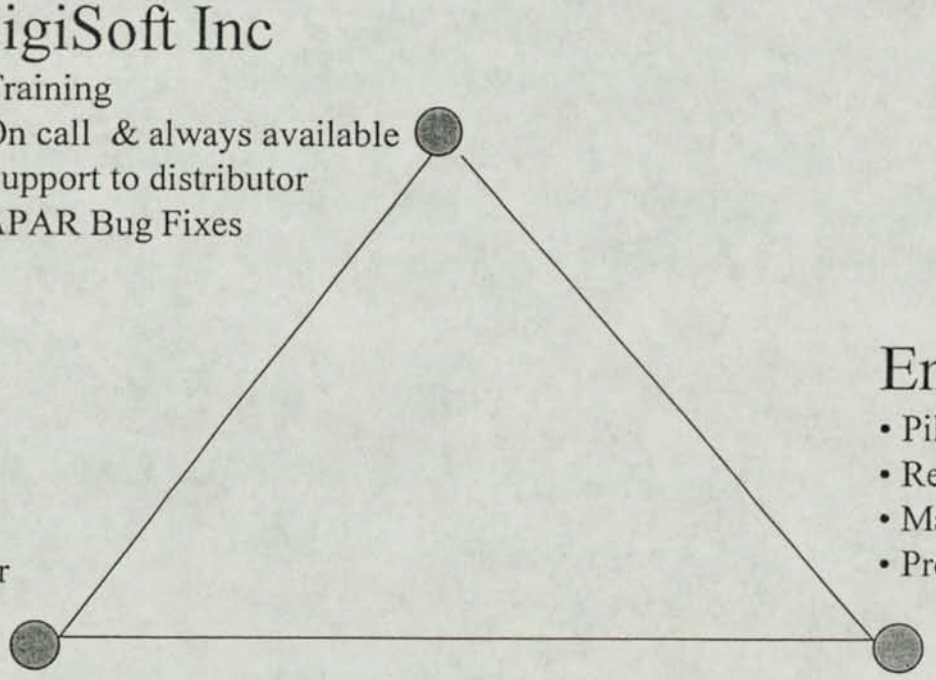
System Test

Action	Who	Est. Time
▪ Run enabled system	End User	1 day
▪ Compare to Baseline	End User	1 day
▪ Run system in future	End User	1 day
▪ Visually inspect results	End User	5 days
Total		8 days

BigiSoft, Inc.

Vertex 2000 Pilot Project Roles

- Based on current BETA experiences



Product Installation

Action	Who	Est. Time
▪ Train participants	Bigisoft	1 day
▪ Plan installation	All	1 day
▪ Install product	End User	1 day
▪ Set up Test Platform	End User	~ 2 day
Total		5 days

Estimated Project Duration

Action	Est. Time
■ Product Installation	5 days
■ System Remediation	21 days
■ System Test	8 days
Total	34 days

Release, Maintain and Support

Action	Who
▪ Release	End User
◆ Normal configuration control	
▪ Maintain	End User
◆ Normal disagnostic process	
▪ Support Levels 1& 2	Distributor
◆ Phone and on-site support for abnormal problems	
▪ Support Level 3	BigiSoft
◆ 24 hour APAR level emergency updates	

Telecom Company BETA site

- Dual Teams - Vertex 2000 v. Y2K Factory Tools
- Project Size
  - 50 COBOL/VS programs - CICS online and batch
  - 12 VSAM master files - 30 reports
- Results
  - V2K 1 man project complete in 3 days
  - Other 3 man team not in test in 10 days
  - 10 X 1 productivity boost minimum

Desired Target Environments

- Platforms
  - MVS/ESA, MVS/XA, All machine types
- Compilers
  - Any IBM COBOL compilers not previously listed
- Data Bases
  - IDMS, DB2, ADABAS, VSAM, or others
- Interfaces
  - Non-3270 BMS terminal devices

What we want from you

- Comments on your impression of the strengths and weaknesses of this product
- Access to new target environments for more testing and evaluation
- Support in implementation of Vertex 2000 Pilot Project at your site
- Assistance in identifying any "must have" before we can launch this product commercially

```
   Menu  Utilities  Compilers  Help
 ------------------------------------------------------------------------------
 BROWSE    BR03.V2K.VDATAO(COMPLEX)                  Line 00000007 Col 001 076
 Command ===>                                          Scroll ===> PAGE
D..9..n......\..J...
C07F829200012E23D012
49790750811180081118
 ------------------------------------------------------------------------------
******************************** Bottom of Data ********************************
```

```
   Menu  Utilities  Compilers  Help
 ------------------------------------------------------------------------------
 BROWSE    BR03.V2K.VDATAI(COMPLEX) - 01.03          Line 00000007 Col 001 076
 Command ===>                                          Scroll ===> PAGE
D.*...n.............
C07082920001202300 12
49F90F50F111F00F111F
 ------------------------------------------------------------------------------
******************************** Bottom of Data ********************************
```