Year 2000 Hex Storage

Kirk Paul Lafler, Millennium Technologies Institute, Inc.
Sunil Kumar Gupta, Millennium Technologies Institute, Inc.
Charles Edwin Shipp, Millennium Technologies Institute, Inc.

ABSTRACT
Legacy systems that store, display or print dates in two-character representation for years ('97, '98, '99, ...) will have an ambiguity problem when any of the year values go beyond the year 1999. Serious time-span calculations and other problems will then occur. The SAS option YEARCUTOFF provides a user-selected 100-year window and is good for many situations. Another solution is to read and write the storage using hex to retain the old record layout. Two hexadecimal bytes can represent year values from 0 to over 65,000, removing the 100-year limit. This method has the advantage of not needing to redesign old file structures, and the added advantage that other languages can also read the files. Several other options are presented.

INTRODUCTION
There are several ways to solve the Year 2000 dilemma and what is best for one system will not be the best for other systems. Each system needs to be considered individually based on the complexity of 1. input/output screens; 2. database; 3. flat files (input/output); 4. program variables and logic; and 5. printed output reports. For systems deemed valuable enough to continue (not be absorbed into another project, etc.) there are several choices available.

This coders corner paper presents several approaches to solve the Year 2000 programming problem.

SOLUTION #1
Your first choice is to expand all dates to four-digit year values. This may be easier than you first think, especially if your dates are stored in SAS files. You could have a system with very little date values involved. Flat files may provide challenges though.

SOLUTION #2
A good solution for the systems that have dates that are rather close (within 25 years before and after) of the current date, then you can feel confident that a 100-year fixed window will include your dates. You can chose a 25/75, 50/50, or 75/25 year window depending on if you feel dates are more prevalent in the future, past, or about equally balanced. Then use the following SAS Options statement at the top of your program:

OPTIONS YEARCUTOFF=1950; for [1950, 2049]
or
OPTIONS YEARCUTOFF=1925; for [1925, 2024]
or
OPTIONS YEARCUTOFF=1975; for [1975, 2074]

This allows you to avoid a potentially large programming effort and still have the services of your computer system. This method is very valuable when things are complex. A very simple statement in your program adjusts all of the standard SAS date results. Note that non-standard handling of year values are not corrected.

SOLUTION #3
Suppose you want your window for dates to be four or five year into the future and 95 years in the past. Plus you want it to move each year with the current year. You can then program a "Sliding 100-Year Window" using the following statements:

*** ENTER WITH YR IN TWO-DIGITS ('99,'00,'01),
*** EXIT WITH YEAR IN FOUR-DIGITS,  

YRTEST1 = YEAR(TODAY())+9-2000;  
IF YR <= YRTEST1 THEN YEAR = 2000 + YR;  
ELSE YEAR = 1900 + YR;

This gives you a sliding window that covers 90 years in the past and 9 years in the future.

SOLUTION #4
Larger problems come when multiple systems rely on the same flat files, and they have two-digit year values. Making one simple change from two-digit to four-digit year values can have a domino effect, and in many instances that approach must be avoided -- it is not an option. In these instances, retaining the relative position and lengths of variable fields is possible by
changing the input/output year formats from numeric (2.) to hexadecimal such as PK2. (packed.) One then needs to consider a planned migration in program version and data version, but it is certainly a viable solution to a difficult problem.

EXAMPLE CODE (BEFORE)
INFILE DD1;
INPUT ID $3. PART $CHAR12. YR1 2. YR2 2. ;

EXAMPLE CODE (AFTER)
INFILE DD1;
INPUT ID $3. PART $CHAR12.
YR1 PK2. YR2 PK2.;

It is a good idea to make two copies of all programs and data involved: one for backup and one for testing. To move entirely in this direction (using PK2. fields) you can have a utility program write over the TEST data sets so they have four-digit dates.

SOLUTION #N
A number of solutions are available commercially and fall into such interesting categories as "time-warping", "encapsulation", and "encryption". Contact the authors of this paper for further details.

SOLUTION # 00
There are cases where you will want to do nothing. If your system has two-digit dates but they are only for display and printing, with no calculations and no possibility of being ambiguous, then an option is to just let them appear as will be common usage. For example MM/DD/YY for Independence Day for the next few years would be: 07/04/97, 07/04/98, 07/04/99, 07/04/00, 07/04/01, 07/04/02, 07/04/03, ... If the values are for display only, then nothing needs to be done. Be careful, however, that there is no sorting, comparisons, nor time-span calculations. In these cases chose another solution.

SUMMARY
Because there are so many data files and programs to fix, each company needs to prioritize the efforts. In addition, it is well to select the solution best suited for each situation. It will not be an option to expand all systems to four-digit years, nor is it the best solution for all systems. Past experience indicates that the YEARCUTOFF window (where the value sets the beginning edge of the 100-year window) is an excellent choice of solution due to the fact that it will save 70% of the reprogramming costs and have the same end results. Further, a sliding window would provide a perpetual solution.

A SAS FREQUENTLY ASKED QUESTION
A wise person once said, "there is no good time to have a problem with your computer or the software you are using." Since the Year 2000 and its associated problems will affect tens of millions, and most probably hundreds of millions, this statement seems most appropriate. As is often the case with complex and/or demanding problems, an array of questions arise. The following Frequently Asked Question/Answer sheds some light for the reader. The reader is advised to look elsewhere (i.e., Internet) for a more detailed and comprehensive list of FAQs.

Q. Will SAS software handle dates in the twenty-first century?
A. Yes. SAS software is already equipped to handle the date transition. If dates in your file consist of a four-digit year, then SAS can accept and display these with little or no difficulty. If dates in your file consist of a two-digit year, then the use of the system option YEARCUTOFF= may be necessary to differentiate the desired century.

CONCLUSION
In prioritizing, companies and institutions will want to look at the total enterprise and consider the options of "re-engineering" which is to say, combining some programs into other programs, raising the priority of the best and reducing the priority of the less important. Those companies that are aware and prepared will have the competitive edge to take them to the next level in the twenty-first century; and the systems that are well-tested will be the ones that survive, to the profit of their owners.

ACKNOWLEDGMENTS
The authors would like to thank Margaret Hung, PharmaSUG Conference Chair, and the other officers, for their support and encouragement during the development of this paper.
Select a Year 2000 Solution
(Written in "pseudo-code")

(1) DO ... EXAMINE your System/Programs/Data;

(2) If the representation of years is OK, and there is no: 1. time-span calculations,
    nor 2. year-sorting,
    nor 3. year-comparisons,
then you could consider leaving it ... "As Is."

(3) If (Priority > Difficulty) then expand all fields;
    * From two-digit to four-digit year fields;
    * in all pgms/screens/data/printed-reports;

(4) Else If
    1. you use standard SAS date functions, and
    2. your dates are close together ( < 100 yrs )
then consider the Fixed Window:
    OPTION YEARCUTOFF=19yy;

(5) Else if you prefer a Sliding Window: use
    %SYSFUNC to execute functions by the macro facility
    such that: OPTION YEARCUTOFF =
    %EVAL(%SYSFUNC(YEAR(%SYSFUNC(TODAY()))) - 90);

(6) With choices (4) and (5), remember also to still check output reports and input/output screens to be to
    your liking (and your customers) i.e. four-digit years.

(7) ELSE IF (Other languages use files) or (years spanned are too wide) then use packed decimal (PK2.)
    for years;
    * will pack four-digit values into two columns;

(8) ELSE IF (for complexity reasons) or (works better for you) then consider vendor solutions;
    * such as Time-Warp, Encapsulation, ........
    * (patented solutions);

(9) Last Resort (sometimes best) is to re-engineer : to
    place program features into other systems -- possibly
    Year 2000 compliant "off the shelf" or turnkey
    software.

Table 1. Decision Tree