CHAPTER

14

Considerations for Y2K and Beyond

Disclaimer

This information is provided to assist customers in evaluating Year 2000 issues and it is subject to change without notice. Its purpose is to provide information only; warranties for SAS software products are contained in the customer's License Agreement with SAS Institute. The information contained here does not constitute an extension of any warranty for SAS software products nor does it guarantee that any software is error-free. The Internet address http://www.sas.com/y2k provides more information and advice.

Definition – What Is the Y2K Problem?

Using dates that indicate only the last two digits of the year can create problems because it may not be certain which century the two-digit years are referring to. For example, the date 1/21/00 may represent January 21, 1900, or January 21, 2000. Computer systems risk generating errors if they fail to distinguish which century such dates refer to.

How SAS Handles Dates

How SAS Stores Date Values

To evaluate your SAS code for Year 2000 compliance (and to understand why SAS software itself is compliant), it is helpful to understand how SAS software processes date values. The SAS System stores date values as an offset in days from January 1, 1960, and it stores datetime values (such as 23FEB98:16:30) as an offset in seconds from midnight January 1, 1960.

Dates and times converted to SAS date and datetime values in this manner may be used more easily in numeric calculations, and there is no ambiguity as to which century a SAS date or datetime value is associated with.

For more information about how the SAS system handles dates, see the section on dates, times and datetime values.

How SAS Converts External Date Information to Internal Representation

You can convert external data to or from SAS date values and datetime values by using SAS informats, formats, and functions. Most of these features accept two-digit years as well as four-digit years. For example, if your input file contains a date value of ’07/27/1998’, you can read it using the MMDDYY10. informat, while a field containing ’07/27/98’ can be read using either the MMDDYY8. informat or the MMDDYY10 informat. Similarly, dates with four and two-digit years can be written to external files using the MMDDYY10. and MMDDYY8. formats respectively.

How to Read Two-Digit Years Using YEARCUTOFF=

If dates in your external data sources or SAS program statements contain two-digit years, you can determine which century prefix should be assigned to them by using the YEARCUTOFF= system option. The YEARCUTOFF= system option specifies the first year of the 100-year span that is used to determine the century of a two-digit year.

Before you use the YEARCUTOFF= system option, examine the dates in your data:

□ If the dates in your data fall within a 100-year span, you can use the YEARCUTOFF= system option.

□ If the dates in your data do not fall within a 100-year span, you must either convert the two-digit years to to four-digit years or use a DATA step with conditional logic to assign the proper century prefix.

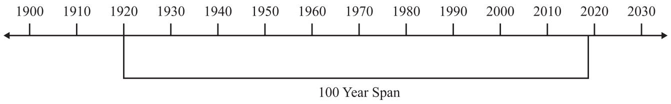
Once you’ve determined that the YEARCUTOFF= system option is appropriate for your range of data, you can determine the setting to use. The best setting for YEARCUTOFF= is a year just slightly lower than the lowest year in your data. For example, if you have data in a range from 1921 to 1999, set YEARCUTOFF= to 1920, if that is not already your system default. The result of setting YEARCUTOFF= to 1920 is that

□ SAS interprets all two-digit dates in the range of 20 through 99 as 1920 through 1999.

□ SAS interprets all two-digit dates in the range 00 through 19 as 2000 through 2019.

The following figure shows the span of years when the YEARCUTOFF= option is set to a value of 1920. The 100-year span in this case is from 1920 to 2019.

Display 14.1 Span of Years When the YEARCUTOFF= Option Is Set to 1920



With YEARCUTOFF= set to 1920, a two-digit year of 10 would be interpreted as 2010, and a two-digit year of 22 would be interpreted as 1922.

Here are some other helpful facts to know when using the YEARCUTOFF= system option:

- □ Although the default value for YEARCUTOFF= is 1920, your system administrator may change the default to suit specific needs at your work site. To determine the current YEARCUTOFF= setting, run PROC OPTIONS:

  ```
  proc options option=yearcutoff;
  run;
  ```

  The current YEARCUTOFF= setting is then displayed in the SAS log.
- □ When you adjust the YEARCUTOFF= system option in a SAS job, the new value only persists for the duration of the SAS session.
- □ You cannot specify more than one value for the YEARCUTOFF= option within a single DATA step.
- □ Valid values for the YEARCUTOFF= option are 1582 to 19900.

Example 1: How YEARCUTOFF= Affects Two and Four-Digit Years

The following example shows what happens with data that contains both two and four-digit years. Note how the YEARCUTOFF= option is set to 1920.

```
options yearcutoff=1920 nodate pageno=1 linesize=80 pagesize=60;

data schedule;
   input @1 jobid $ @6 projdate mmddyy10.;
   datalines;
A100 01/15/25
A110 03/15/2025
A200 01/30/96
B100 02/05/00
B200 06/15/2000
 ;

proc print data=schedule;
   format projdate mmddyy10.;
run;
```

The resulting output from the PROC PRINT statement looks like this:

Output 14.1 Output from The Previous DATA Step Showing 4–Digit Years That Result from Setting YEARCUTOFF= to 1920

```
                              The SAS System         1

                        Obs    jobid      projdate

                         1     A100     01/15/1925
                         2     A110     03/15/2025
                         3     A200     01/30/1996
                         4     B100     02/05/2000
                         5     B200     06/15/2000
```

Here are some facts to note in this example:

- □ In the datalines in the DATA step, the first record contains a two-digit year of 25, and the second record contains a four-digit year of 2025. Because the YEARCUTOFF= system option is set to 1920, the two-digit year defaults to a year in the 1900s in observation number 1. The four-digit year in observation number 2 is unaffected by the YEARCUTOFF= option.
- □ The third record is similar to the first and defaults to a year in the 1900s based on the value of YEARCUTOFF=.
- □ The output from records 4 and 5 shows results that are similar to records 1 and 2. The fourth record specifies a two-digit year of 00, and the fifth one specifies a four-digit year of 2000. Because of the value of the YEARCUTOFF= option, the years in the two resulting observations are the same.

As you can see, specifying a two-digit year may or may not result in the intended century prefix. The optimal value of the YEARCUTOFF= option depends on the range of the dates that you are processing.

In Releases 6.06 through 6.12 of the SAS System, the default value for the YEARCUTOFF= system option is 1900; in Version 7 and Version 8, the default value is 1920.

For more information on how SAS handles dates, see the section on dates, times and datetime values.

Possible Year 2000 Problems and Strategies

Summary of Potential Problem Areas

It is important to be aware of all aspects of your data processing environment that may cause misinterpretation of dates. This includes all files and programs in your SAS libraries, as well as files and raw data from external sources. The following list contains potential problem areas:

- □ *Input data screens* may contain two-digit date fields to save keystrokes.
- □ *SAS programs*
  - □ may write dates with two-digit years to output such as reports and external files
  - □ may misinterpret two-digit years in input data
  - □ may misinterpret two-digit years in program code elements such as date constants or arguments to date functions.

- □ *SAS datasets* may contain two-digit years stored in character or non-SAS date or datetime numeric variables.
- □ *Raw data* may contain two-digit years.
- □ *Display or print output headers* may display two-digit years.

Practices That Help Insure Date Integrity

The following practices help ensure that your date values are correct during all the conversions that occur during processing:

- □ Store dates as SAS date values, not as simple numeric or character values.
- □ Use the YEARCUTOFF= system option when converting two-digit dates to SAS date values.
- □ Examine sets of raw data coming into your SAS process to make sure that any dates containing two-digit years will be correctly interpreted by the YEARCUTOFF= system option. Look out for
  - □ two-digit years that are distributed over more than a 100-year period. For dates covering more than a 100-year span, you must either use four digit years in the data, or use conditional logic in a DATA step to interpret them correctly.
  - □ two-digit years that need an adjustment to the default YEARCUTOFF= range. For example, if the default value for YEARCUTOFF= in your operating environment is 1920 and you have a two-digit date in your data that represents 1919, you will have to adjust your YEARCUTOFF= value downward by a year in the SAS program that processes this value.
- □ Make sure that output SAS data sets represent dates as SAS date values.
- □ Check your SAS programs to make sure that formats and informats that use two-digit years, such as DATE7., MMDDYY6., or MMDDYY8., are reading and writing data correctly.

Note: The YEARCUTOFF= option has no effect on dates that are already stored as SAS date values. △

Tools That Help Ensure Data Integrity

Here is a summary of the SAS System tools you'll be using to insure the integrity of your date-related data.

YEARCUTOFF= SAS system option
specifies the proper century to associate with two-digit years.

Note: A Technical Tip on the YEARCUTOFF= option is available from the SAS web site: http://www.sas.com/techsup/download/technote/ts597.html. △

informats, formats and functions
are SAS System language elements that handle a wide variety of reading, writing and formatting operations related to dates and time, such as

- □ converting dates from one format to another, such as mmddyy to mmddyyyy
- □ converting dates to SAS date values, so that numeric operations may be performed on them
- □ converting SAS date or datetime values back to a variety of date or time formats, such as 062099, or June 20, 1999.

Strategies for Fixing Problems

Not all potential problems need to be corrected. If your SAS program writes out date values with two-digit years, you may not need to modify the program if the year can be correctly inferred from the context. For example, a printed report listing expiration dates for current drivers licenses can use two-digit years. However, if the same program writes out the dates to a file, and a subsequent application reads the two-digit year and subtracts it from another two-digit year, then the program(s) should be changed. Here are some (but by no means all) steps you can take to correct potential compliance problems.

- □ Make sure that an appropriate value for the YEARCUTOFF= option is specified in your application. The specific value to use will obviously depend on the range of dates your applications use. Note that if all of your applications use the same general range, you can set the YEARCUTOFF= option at the system level so that all applications use the same default. Note also that you can specify the option at any point in your program to adjust for different date ranges in different input files.
- □ Make sure all date information is stored as SAS date variables. Various functions are available to convert date information from other formats to SAS date values. This may also require modifying subsequent statements in your program to process the values as date values, but doing this will save you trouble in the long run.
- □ Change instances of two-digit year date formats (such as DATE7.) to four-digit years (such as DATE9.). This may not always be practical, particularly when you write output files, since the format of the file (and any applications that process it) may need to be modified to adjust for the extra digits.
- □ Modify applications that input date information through screens or forms to require or allow four-digit years.
- □ If variables in SAS data sets are assigned permanent formats, use PROC DATASETS or a DATA step to change the formats to ones using four-digit years.
- □ Assign appropriate formats and controls to date fields that are imported through SAS/ACCESS interfaces.

Examples: Fixing Problems

Example 2: Reading Two-Digit Year Values in Raw Data

This example demonstrates how to use the YEARCUTOFF= system option to read a raw data file that contains two-digit year values. Here are the input data records:

```
ADAMS,GERALD,M,819-68-375,01MAY99
AVERY,JERRY,M,944-69-923,01MAY02
BLALOCK,RALPH,M,189-62-924,01MAY02
BRADLEY,JEREMY,M,142-96-693,01MAY00
BURNETTE,THOMAS,M,922-64-294,01MAY99
CARTER,DOROTHY,F,441-87-333,01MAY00
CHIN,JACK,M,438-82-579,01MAY99
DAVIDSON,JASON,M,617-00-610,01MAY99
```

```
   DENNIS,ROGER,M,342-14-341,01MAY99
   EDGERTON,JOSHUA,M,778-62-406,01MAY00
```

In the DATA step, the YEARCUTOFF= system option is set to 1900 and produces unintended graduation dates.

```
options YEARCUTOFF=1900 nodate pageno=1 linesize=80 pagesize=60;
data students;
    infile 'students-data-file' dlm=',';
    input lname: $15. fname: $15. gender: $1. idnum: $11.
          graddate: date7.;
    format graddate date9.;
run;

proc print data=students noobs label;
    title1 'Projected Graduation Dates';
    title2 'Report Date: Sep 30, 1999';
    label lname='Last Name'
          fname='First Name'
          gender='Gender'
          idnum='Student Id'
          graddate='Projected Graduation';
run;
```

The following output shows the resulting dates in the last column of the report.

Output 14.2 Output Data Set Showing Errors Due to Incorrect YEARCUTOFF= Value of 1900

```
                 Projected Graduation Dates              1
                     Report Date: Sept 30, 1999

  Last        First                               Projected
  Name        Name       Gender    Student Id    Graduation

  ADAMS       GERALD       M       819-68-375    01MAY1999
  AVERY       JERRY        M       944-69-923    01MAY1999
  BLALOCK     RALPH        M       189-62-924    01MAY1902
  BRADLEY     JEREMY       M       142-96-693    01MAY1900
  BURNETTE    THOMAS       M       922-64-294    01MAY1999
  CARTER      DOROTHY      F       441-87-333    01MAY1900
  CHIN        JACK         M       438-82-579    01MAY1999
  DAVIDSON    JASON        M       617-00-610    01MAY1999
  DENNIS      ROGER        M       342-14-341    01MAY1999
  EDGERTON    JOSHUA       M       778-62-406    01MAY1900
```

To correct the problem in the resulting dates, specify the appropriate YEARCUTOFF= value and rerun the DATA step.

```
options YEARCUTOFF=1950;
```

This next output results from the same DATA step program with the OPTIONS statement containing a YEARCUTOFF= value set to 1950. Note that for the raw data shown in this example, the YEARCUTOFF= system option will yield correct results when set within a range of 1903 to 1998.

Output 14.3 Output Data Set Showing Corrected Date Values Due to Setting the YEARCUTOFF= Value to 1950

```
                  Projected Graduation Dates                  1
                      Report Date: Sept 30, 1999

Last        First                                Projected
Name        Name       Gender    Student Id     Graduation

ADAMS       GERALD       M       819-68-375     01MAY1999
AVERY       JERRY        M       944-69-923     01MAY1999
BLALOCK     RALPH        M       189-62-924     01MAY2002
BRADLEY     JEREMY       M       142-96-693     01MAY2000
BURNETTE    THOMAS       M       922-64-294     01MAY1999
CARTER      DOROTHY      F       441-87-333     01MAY2000
CHIN        JACK         M       438-82-579     01MAY1999
DAVIDSON    JASON        M       617-00-610     01MAY1999
DENNIS      ROGER        M       342-14-341     01MAY1999
EDGERTON    JOSHUA       M       778-62-406     01MAY2000
```

Note: The new value for YEARCUTOFF remains in effect for the remainder of the SAS job or SAS session, or until it is changed again. △

Example 3: Expanding Dates in External Files

When you make structural changes to stored data, you must modify all applications that access the changed data in order to accommodate the new data format or data definition. It may be less disruptive to add expanded fields to the end of the record and leave the rest unchanged.

If you decide you must alter the contents of an external file, you can use a DATA step to modify or correct the stored data. Changing two-digit years to four-digit years in raw data files by adding the century is called field expansion. Additional disk space is required to store the expanded data.

In the following example, flight dates are stored in a mmddyy format (030195) in a raw data file in columns 4-9. We'll expand the date field from 6 to 8 columns by inserting the century values (03011995).

Here's a partial listing of the original raw data file:

```
----|----10---|----20---|----30---|----40---|----50
182030195 8:21LGAYYZ  366 458 390104 16  3123178
114030195 7:10LGALAX2,475 357 390172 18  6196210
20203019510:43LGAORD  740 369 244151 11  5157210
219030195 9:31LGAFRA3,442 412 334198 17  7222250
43903019512:16LGALAX2,475 422 267167 13  5185210
38703019511:40LGACPH3,856 423 398152  8  3163250
290030195 6:56LGAWAS  229 327 253 96 16  7117180
52303019515:19LGAORD  740 476 456177 20  3185210
98203019510:28LGADFW1,383 383 355 49 19  2 56180
62203019512:19LGALON3,857 255 243207 15  5227250
```

The following program demonstrates field-expansion:

```
options yearcutoff=1950 nodate pageno=1 linesize=80 pagesize=60;
   data _null_;
     infile 'march.dat' truncover;
     input @1 string1 $char3.
           @4 date mmddyy6.
           @10 string2 $char40.;
     file 'march.dat';
     put   @1 string 1 $char3.
           @4 date mmddyyn8.
           @12 string2 $char40.;
run;
```

Here is a partial listing of MARCH2.DAT after expanding the two-digit year values to four digits:

```
----|----10---|----20---|----30---|----40---|----50
18203011995 8:21LGAYYZ  366 458 390104 16  3123178
11403011995 7:10LGALAX2,475 357 390172 18  6196210
2020301199510:43LGAORD  740 369 244151 11  5157210
21903011995 9:31LGAFRA3,442 412 334198 17  7222250
4390301199512:16LGALAX2,475 422 267167 13  5185210
3870301199511:40LGACPH3,856 423 398152  8  3163250
29003011995 6:56LGAWAS  229 327 253 96 16  7117180
5230301199515:19LGAORD  740 476 456177 20  3185210
9820301199510:28LGADFW1,383 383 355 49 19  2 56180
6220301199512:19LGALON3,857 255 243207 15  5227250
```

Example 4: Converting Dates Stored As Character Values to SAS Date Values

This example converts dates in character variables in a SAS data set, to SAS date values in a SAS data set. The Birth dates in the SAS data set YR20.SENIORS are stored in the character variable BIRTHDATE using a DDMMMYYYY format (18SEP1898). Some records contain two-digit years, others use four-digit years:

```
Listing of Seniors Birth Dates
     LNAME              FNAME       GENDER     BIRTHDATE         ID
    ADAMS             GERALD          M       04DEC96      259398646
    AVERY             JERRY           M       27JAN25      531691723
    BLALOCK           RALPH           M       06APR1888    523870522
    BRADLEY           JEREMY          M       26SEP1922    297193965
    BURNETTE          THOMAS          M       16FEB1921    226507519
    CARTER            DOROTHY         F       05APR18      287225611
    CHIN              JACK            M       24APR19      590364670
    DAVIDSON          JASON           M       16FEB1918    506603530
    DENNIS            ROGER           M       11MAR26      297222847
    EDGERTON          JOSHUA          M       10JAN06      168088351
```

The following SAS program uses the INPUT function to convert the character dates to SAS date values to enable chronological sequencing in a report. Because some birth dates are before 1900, note that the YEARCUTOFF= value must be adjusted.

```
options yearcutoff=1880;
options nodate pageno=1 linesize=80 pagesize=60;
data seniors(drop=tempvar);
   set yr20_seniors(rename=(Birthdate=tempvar));
   Birthdate=input(tempvar,date9.);
run;
```

Note: The DATE9. informat correctly reads two-digit as well as four-digit years. △

The following program illustrates how to print the WORK.SENIORS data set. The birth dates, in the BIRTHDATE variable, are written with a DATE9. format.

```
proc print data=seniors noobs;
   format birthdate date9. ID ssn.;
   title 'Listing of Seniors Birth Dates';
run;
```

The PROC PRINT step produces the listing of the reformatted WORK.SENIORS data set:

Output 14.4 Reformatted Data Set WORK.SENIORS

```
                       Listing of Seniors Birth Dates                      1

           Lastname    Firstname    Gender        SSN        Birthdate

           ADAMS        GERALD        M       259-39-8646    04DEC1896
           AVERY        JERRY         M       531-69-1723    27JAN1925
           BLALOCK      RALPH         M       523-87-0522    06APR1888
           BRADLEY      JEREMY        M       297-19-3965    26SEP1922
           BURNETTE     THOMAS        M       226-50-7519    16FEB1921
           CARTER       DOROTHY       F       287-22-5611    05APR1918
           CHIN         JACK          M       590-36-4670    24APR1919
           DAVIDSON     JASON         M       506-60-3530    16FEB1918
           DENNIS       ROGER         M       297-22-2847    11MAR1926
           EDGERTON     JOSHUA        M       168-08-8351    10JAN1906
```

Example 5: Converting Data Stored As Simple Numeric Values To SAS Date Values

This example shows how the SAS System recognizes date values, which are stored as simple numeric variables, and displays them as more familiar date values in SAS data sets. Even though SAS is able to present the date values in many formats, the SAS system stores each date as a SAS date, which is a value equal to the number of days between January 1, 1960 and the calendar date. In the following SAS data set YR20.JUNIORS birth dates are stored as numeric variables in Julian (BDJULIAN) and numeric YYYYMMDD format (BDCOBOL). Note that in the first observation, the number 2003116 is a Julian date that represents the 116th day of year 2003. This date will be recognized by the SAS System, stored as a SAS date, and displayed in a more familiar DDMMMYY date format. Here is a partial listing of the YR20.JUNIORS data set:

```
Last Name     First Name   Gender     BDJULIAN   BDCOBOL
ALEXANDER     SUSAN          F        2003116    20030426
BAREFOOT      JOSEPH         M        1997103    19970413
BOSTIC        MARIE          F        1998143    19980523
BRADY         CHRISTINE      F        2000289    20001015
CAHILL        MARSHALL       M        2004305    20041031
CARTER        KAREN          F        1999068    19990309
```

```
CHOW          JANE           F        1998293    19981020
DEAN          SANDRA         F        1997111    19970421
DONALDSON     KAREN          F        1999030    19990130
FERNANDEZ     KATRINA        F        1997042    19970211
FOSTER        GERALD         M        1999213    19990801
GRAHAM        ALVIN          M        1997320    19971116
HARRISON      FELICIA        F        1999147    19990527
HOWARD        GRETCHEN       F        1999298    19991025
```

The following program example takes the numeric date values, converts the Julian dates with the DATEJUL function and converts the COBOL dates with the PUT and INPUT functions, and then prints the results.

```
options nodate pageno=1 linesize=80 pagesize=60 yearcutoff=1960;

data juniors;
   set yr20_juniors;
   sasdate1=datejul(bdjulian);
   sasdate2=input(put(bdcobol,8.),yymmdd8.);
run;

title 'Birth Dates converted to SAS dates';
proc print data=juniors (obs=14) noobs;
   format sasdate1 sasdate2 date9.;
run;
```

The following output displays the first 14 observations in the output data set:

Output 14.5 Output Showing Converted Dates from YR20.JUNIORS Data Set

```
                     Birth Dates converted to SAS dates                    1

 Lname        Fname        Gender   bdjulian    bdcobol    sasdate1    sasdate2

 ALEXANDER    SUSAN          F       2003116   20030426   26APR2003   26APR2003
 BAREFOOT     JOSEPH         M       1997103   19970413   13APR1997   13APR1997
 BOSTIC       MARIE          F       1998143   19980523   23MAY1998   23MAY1998
 BRADY        CHRISTINE      F       2000289   20001015   15OCT2000   15OCT2000
 CAHILL       MARSHALL       M       2004305   20041031   31OCT2004   31OCT2004
 CARTER       KAREN          F       1999068   19990309   09MAR1999   09MAR1999
 CHOW         JANE           F       1998293   19981020   20OCT1998   20OCT1998
 DEAN         SANDRA         F       1997111   19970421   21APR1997   21APR1997
 DONALDSON    KAREN          F       1999030   19990130   30JAN1999   30JAN1999
 FERNANDEZ    KATRINA        F       1997042   19970211   11FEB1997   11FEB1997
 FOSTER       GERALD         M       1999213   19990801   01AUG1999   01AUG1999
 GRAHAM       ALVIN          M       1997320   19971116   16NOV1997   16NOV1997
 HARRISON     FELICIA        F       1999147   19990527   27MAY1999   27MAY1999
 HOWARD       GRETCHEN       F       1999298   19991025   25OCT1999   25OCT1999
```

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS Language Reference: Concepts*, Cary, NC: SAS Institute Inc., 1999. 554 pages.

SAS Language Reference: Concepts

ISBN 1–58025–441–1

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, November 1999

The Institute is a private company devoted to the support and further development of its software and related services.