

SAS and the Year 2000 Problem

Clarence Wm. Jackson
City of Dallas, Texas

The purpose of this paper is to review the Year 2000 Problem, and some tools available within the SAS System. SAS System tools such as the YEARCUTOFF system option, various date time informats, formats, and functions will be reviewed.

Introduction

The significance of the Year 2000 problem on SAS programs and other computer systems vary depending on your point of view, but it will affect each of us in some fashion. The root of and reason for the problem is very simple to understand: no century digits in date fields. As anyone will tell you who was programming back in the early days, two digits was worth a lot of resources. When everything was expensive (memory and storage space), or limited (memory, storage space, ALC program sizes, paper cards), saving space was saving money. Two digits multiplied by the total records could save a lot of resources and money depending of the size of your data and process.

The early pioneers of programming dropped the century digits to save money on the scarce resources. Why waste resources to process "19" in calculations when it wasn't "significant" to the results? This practice was taken further into newer software and hardware applications, from BIOS to networks, MVS to PC's, cash machines, telephone systems, and most other software and hardware that use dates as part of it's function.

The solution then to conserve the resources has resulted in the problem we face now. The century data was not being stored, and this practice spread like a virus, being taught to new programmers as the way to save resources. Other "bad" programming techniques related to date processing were also born during these days, such as using "00" and "99" in year calculations and high-low range logic tests, and hard coded "19" century values in data presentations such as reports and screen displays. Then there are the record keys based on dates, used to retain the proper sequence in the records.

As the cost of resources dropped, the practice didn't, but were put into the new resources, products, and applications to have "compatibility" with older systems, programs and files. So the two digits were never put into many of the systems in use today, and that is why there will be a panic in the next few years as some systems will start to fail because of the dates.

I use SAS, so I Don't Care!

As SAS System Software Users, we may feel that we are not going to be affected, as there are many available options to working with dates in many input formats. Even with the 2 digit year, SAS can process the data by modifying the YEARCUTOFF system option from '1900' to whatever date you want. SAS can read the date data and correctly process dates from 1582 A.D. to 20,000 A.D., and print it too! However, there is still reasons for concerns. Listed are two main areas of concern that SAS programmers should be aware of:

- processing raw data from non SAS data sources, and
- hardware timers.

There will be changes made to address both concerns as date fields finally are converted to include the century digits, off-the-shelf applications are made Year 2000 compliant, and hardware makers upgrade their products. Since the hardware timers feed the applications the system's time as stored, look for problems with data coming from older machines, PC or mainframe. The date processing problems will cause a lot of changes to the raw data that you'll see as a lot of files are converted before the big day.

Raw Data - date representation, it's structure and content is the most overriding factor. The structure of the date must include the century or century value as part of the date. Date must be converted if this structure is

not currently in place, causing files to change. If the dates are not converted, then some type of software logic must be used to "work around" this. Dates that look like MMDDYY or any variant will need to be changed to look like MMDDCCYY, where CC=Century. The source of data will also become a problem, as imported data and their systems may not support the century date. SAS will be able to read the data, but a lot of software won't, and will require changes in the files thru conversions.

Hardware Timers - hardware considerations boil down to clocks and timers used to feed the software. The mainframe will process years past 1999 (or so we're told), but most personal computer workstations (at least those on the market today) will not because of the system clock only holding the last 2 digits of the year. This will cause problems that no amount of data fixing can solve, and will need to be addressed using software logic to "adjust" for the timers. You can test this on PC's by setting the system clock to December 31, 1999, time 12:58, save a text file to disk, wait until the clock changes to January 1, then save another text file to disk. On Windows, it will show the second text file as being saved in 1900. Networks will require that all time links are updated.

As SAS Users, we don't have to be too concerned with the SAS software. However, the following should be considered when using data from other software. Software applications that process a date field but the software doesn't support the century must be modified, replaced, or worked-around. Special care should be used in any date validation routines that check for year=00 or year=99, as these will be valid years for 1999 and 2000. All software must at least be reviewed, then modified depending on the date data and/or data processing.

On the personal computer software side, there are many major packages that do not handle the year 2000 correctly. Check your software by creating a date field with the value of December 31, 1999, and add '1' (for 1 day) to this value. If you don't come up with the correct date, guess what?

Other Year 2000 Issues that will affect the data you'll be getting from non-SAS sources include, but not limited to:

- a. Arithmetic used in computations of time spans, due dates, other "trigger" dates. Make sure your data source is giving you all the data.
- b. Sorting of data based on dates. Look out for 2000 records coming in sequence before 1999. Remember, the logic is being done (in most shops) using the last 2 digits of the year.
- c. Selections based on date fields that are keys. Could be a Big problem
- d. Backup systems that uses date logic to do incremental backups
- e. Use of 00 and 99 as default values for low and high values. Look for data that may be considered "invalid" before. These are soon going to be good values.
- f. Hard coded 19 in 4 digit year field, and the separate manipulations of those digits.
- g. Automated systems that use dates as part of the automation process to feed your program data.
- h. Screen interfaces, reports, etc
- i. Machine-machine interface via networks
- j. Log files, audit recordings, etc (those SMF and RMF records should really be reviewed)

Steps to minimize the effects of the Year 2000 are simple, especially for SAS programmers, but can become complex for other software programmers when you consider the scope of the changes needed. However, the sooner these changes are made, the less chance of system failures. Things you can do to help you and your company get by:

- a. Do analysis and inventory of date data occurrences in your data, and determine the complexity of the required changes. Review programs and applications that are using the files, update and modify.
- b. Develop a detailed plan for corrections, methods of testing and auditing your conversion process.
- c. Correct the date fields, reformat the files with the expanded layouts, update all data dictionaries and record layouts. You may want to do this off-line and verify results. Then run the corrections against

- the real data when.
- d. Modify programs and applications that are using the files during the same time you are converting the data. This is where SAS programming pays off versus COBOL or ALC on the mainframe.
 - e. Check hardware timers, especially if you are on a network.
 - d. Read up on the SAS System Option **YEARCUTOFF=**. This option is used to set a 100 year sliding window for processing data thru date Informats, Formats, and Functions. The default is set for "1900". Adjust the default as needed for your processing situation. However, if you are processing dates outside the 100 year range of the cutoff date, you may have problems in them be read incorrectly. **Know your data!** To find out what your YEARCUTOFF= value is, run PROC OPTIONS, and look for it under the PORTABLE OPTIONS group.
 - f. Review all date formats and informats used in your programs, and expand the ranges. Although there are certain formats documented in the manuals as having a maximum output length (or range) of eight, you can use a length of ten or more. The more popular formats used are MMDDYYw, YYMMDDw, DDMMYYw. According to the manual, these formats will not support displaying of the century digits (range or w of 10 or more), but the documentation on this point isn't totally correct. I was unable to find the documentation supporting the expanded range, but SAS does support this. (This was an issue with a superior who just read the SAS Language Guide, and hadn't used the product. A copy of the program I used to prove my point is attachment A.)
 - g. Use SAS dates when doing sorts, conditional processing, and setting keys for records. Since SAS uses a serial number internally for the dates (SAS Dates), this will ensure that everything is chronologically correct. Convert all date fields on input to a SAS date before doing anything. Use SAS functions to reformat the data if needed. Depending on your data, if you need to add the century digits in your DATA step by concatenation, then use some functions and format/informats like **INPUT((monvar||"||"||dayvar||"/19"||yearvar),MMDDYY10.)**, or some variation to get your data into shape.

As SAS programmers, there's not the same type of rush to fix this problem, but we still need to be concerned about this problem. Fiscal Year 99/00 is less than 3 years away. If you haven't started your Year 2000 corrections, here's a quick Plan of action:

- define list of items to be changed
- define expanded files
- review changes to be made to the file layouts, increased file sizes
- convert files offline
- review converted files offline using PROC PRINT
- review programs' code for changes to fields being changed by new record layouts
- convert programs' code offline and review
- test against the converted files offline and review results
- change files, programs and retest as needed offline
- final testing and review
- load new files online, programs, etc
- review files, program executions
- review End Of Year ,End Of Month, year 1999 and 2000 processes

Conclusion

Although the problem of the Year 2000 is really about two digits representing the century part of the date, it will have far reaching implications. Although it's really a computer problem, it will affect the businesses that depend on the computers. And in shops where SAS is used mostly by the Data Managers and non-Programmers to read the enterprise data, this may cause serious problems in data. Every system should be reviewed, be it hardware, software, or contains a computer chip, as they are suspect and subject to failure. Although SAS will afford some protection and flexibility to those that are SAS shops, unless we are processing only in-house SAS datasets, we should be aware of what is going on regarding the Year 2000 problem.

Until there are mandatory standards that everyone must follow (in government or not), there will be many

different "solutions" used to fix this problem. The only fix that makes any sense is that we cleanup our act at last, and add the two digits, and adopt the date standards of either ISO, ASCII, or some other standard. Everyone must do the same thing, or we'll spend a lot of time adjusting dates in programs based on the data input sources, which includes other programs, vendors, and others. What good is it to clean it up, then have it corrupted by incoming data from outside your control?

This Year 2000 problem will cause us headaches and will cost us additional resources, but this time we really don't have a choice, and we can't put off the deadline. We need to find our exposure, fix the problems, and get back to work. Or, we can ignore the problem, wait until the problem becomes a system failure, and cause our organizations to go out of business because they can't do business. And then fix the problem under pressure.

ACKNOWLEDGMENTS

- SAS Language Guide, Ver 6, 1st edition, SAS Inst
- SAS Communications, 2nd Quarter 1996 Edition
- The Year 2000 Web Site (WWW.YEAR2000.COM)
- DATAMATION, January 1996
- IBM, THE YEAR 2000 AND 2-DIGIT DATES: A GUIDE FOR PLANNING AND IMPLEMENTATION THIRD EDITION, MAY 1996 (GC28-1251-02)

Author Contact Info:
Clarence Wm. Jackson
73021.3404@compuserve.com (E-mail)
<http://ourworld.compuserve.com/homepages/CJac>
City of Dallas, City Hall 4BS, Dallas Tx 75201
214-670-3704 (voice)

Attachment A - SAS program to Display Years in Various Formats

NOTE: Copyright (c) 1989-1995 by SAS Institute Inc., Cary, NC, USA.

NOTE: SAS (r) Proprietary Software Release 6.11 TS020

Licensed to CITY OF DALLAS, Site 0001457002.

NOTE: AUTOEXEC processing beginning; file is D:\SAS\AUTOEXEC.SAS.

Location of sample information not specified

NOTE: Libref TRANAREA was successfully assigned as follows:

Engine: V606

Physical Name: D:\CJ-ON-D\SASCPORT

NOTE: Libref MYLIB was successfully assigned as follows:

Engine: V611

Physical Name: D:\CJ-ON-D\SASFILES

NOTE: AUTOEXEC processing completed.

```
1
2
3 %let cj=%str(LINK PRINT_IT);
4
5 title1;
6 title2 "&sysjobid, Year 2000 Test for SAS Formats (Using OS/2)";
7
8 DATA YR2000;
9   YEARDATE="01JAN2000"d;
10  &CJ;
11  YEARDATE="01JAN00"d;
12  &CJ;
13  YEARDATE="29FEB96"d;
14  &CJ;
15  YEARDATE="01JAN1900"d;
16  &CJ;
17  YEARDATE="14MAR55"d;
18  &CJ;
19  YEARDATE="01JAN1901"d;
20  &CJ;
21  YEARDATE="01JAN60"d;
22  &CJ;
23  YEARDATE=today();
24  &CJ;
25  YEARDATE="29FEB2000"d;
26  &CJ;
27  YEARDATE="29FEB1996"d;
28  &CJ;
29  YEARDATE=today();
30  &CJ;
31  DO j=1 TO 30;
32    YEARDATE=INTNX('YEAR',YEARDATE,1);
33    &CJ;
34  END;
```

```

35 PRINT _IT:
36 file print;
37 PUT /@1 "_____."
38 /@10 YEARDATE=
39 /@10 YEARDATE MMDDYY8. +1 " AS MMDDYY8."
40 /@10 YEARDATE YYMMDD8. +1 " AS YYMMDD8."
41 /@10 YEARDATE WORDDATE25. +1 " AS WORDDATE25."
42 /@10 YEARDATE WEEKDATE29. +1 " AS WEEKDATE29."
43 /@10 YEARDATE DATE9. +1 " AS DATE9."
44 /@10 YEARDATE MMDDYY10. +1 " AS MMDDYY10."
45 /@10 YEARDATE YYMMDD10. +1 " AS YYMMDD10." ;
46 output;
47 RETURN;
48 RUN;

```

NOTE: 420 lines were written to file PRINT.

NOTE: The data set WORK.YR2000 has 42 observations and 2 variables.

NOTE: The DATA statement used 4.94 seconds.

```

49 proc print n u;
50 format yeardate mmddyy10.;
51 run;

```

NOTE: The PROCEDURE PRINT used 1.03 seconds.

Attachment B - Output from SAS Program

0000000016, Year 2000 Test for SAS Formats (Using OS/2)

YEARDATE=14610
01/01/00 AS MMDDYY8.
00-01-01 AS YYMMDD8.
 January 1, 2000 AS WORDDATE25.
 Saturday, January 1, 2000 AS WEEKDATE29.
01JAN2000 AS DATE9.
01/01/2000 AS MMDDYY10.
2000-01-01 AS YYMMDD10.

YEARDATE=-21914
01/01/00 AS MMDDYY8.
00-01-01 AS YYMMDD8.
 January 1, 1900 AS WORDDATE25.
 Monday, January 1, 1900 AS WEEKDATE29.
01JAN1900 AS DATE9.
01/01/1900 AS MMDDYY10.
1900-01-01 AS YYMMDD10.

YEARDATE=13208
02/29/96 AS MMDDYY8.
96-02-29 AS YYMMDD8.
 February 29, 1996 AS WORDDATE25.
 Thursday, February 29, 1996 AS WEEKDATE29.
29FEB1996 AS DATE9.
02/29/1996 AS MMDDYY10.
1996-02-29 AS YYMMDD10.

YEARDATE=-21914
01/01/00 AS MMDDYY8.
00-01-01 AS YYMMDD8.
 January 1, 1900 AS WORDDATE25.
 Monday, January 1, 1900 AS WEEKDATE29.
01JAN1900 AS DATE9.
01/01/1900 AS MMDDYY10.
1900-01-01 AS YYMMDD10.

YEARDATE=-1754
03/14/55 AS MMDDYY8.
55-03-14 AS YYMMDD8.
 March 14, 1955 AS WORDDATE25.
 Monday, March 14, 1955 AS WEEKDATE29.
14MAR1955 AS DATE9.
03/14/1955 AS MMDDYY10.
1955-03-14 AS YYMMDD10.

YEARDATE=-21549

01/01/01 AS MMDDYY8.
01-01-01 AS YYMMDD8.
January 1, 1901 AS WORDDATE25.
Tuesday, January 1, 1901 AS WEEKDATE29.
01JAN1901 AS DATE9.
01/01/1901 AS MMDDYY10.
1901-01-01 AS YYMMDD10.

YEARDATE=0
01/01/60 AS MMDDYY8.
11:19 Monday, August 12, 1996 2
0000000016, Year 2000 Test for SAS Formats (Using OS/2)
60-01-01 AS YYMMDD8.
January 1, 1960 AS WORDDATE25.
Friday, January 1, 1960 AS WEEKDATE29.
01JAN1960 AS DATE9.
01/01/1960 AS MMDDYY10.
1960-01-01 AS YYMMDD10.

YEARDATE=13373
08/12/96 AS MMDDYY8.
96-08-12 AS YYMMDD8.
August 12, 1996 AS WORDDATE25.
Monday, August 12, 1996 AS WEEKDATE29.
12AUG1996 AS DATE9.
08/12/1996 AS MMDDYY10.
1996-08-12 AS YYMMDD10.

YEARDATE=14669
02/29/00 AS MMDDYY8.
00-02-29 AS YYMMDD8.
February 29, 2000 AS WORDDATE25.
Tuesday, February 29, 2000 AS WEEKDATE29.
29FEB2000 AS DATE9.
02/29/2000 AS MMDDYY10.
2000-02-29 AS YYMMDD10.

YEARDATE=13208
02/29/96 AS MMDDYY8.
96-02-29 AS YYMMDD8.
February 29, 1996 AS WORDDATE25.
Thursday, February 29, 1996 AS WEEKDATE29.
29FEB1996 AS DATE9.
02/29/1996 AS MMDDYY10.
1996-02-29 AS YYMMDD10.

(OTHER DATES DELETED FOR SPACE)

YEARDATE=14245
01/01/99 AS MMDDYY8.
99-01-01 AS YYMMDD8.

January 1, 1999 AS WORDDATE25.
Friday, January 1, 1999 AS WEEKDATE29.
01JAN1999 AS DATE9.
01/01/1999 AS MMDDYY10.
1999-01-01 AS YYMMDD10.

YEARDATE=14610
01/01/00 AS MMDDYY8.
00-01-01 AS YYMMDD8.

January 1, 2000 AS WORDDATE25.
Saturday, January 1, 2000 AS WEEKDATE29.
01JAN2000 AS DATE9.
01/01/2000 AS MMDDYY10.
2000-01-01 AS YYMMDD10.

YEARDATE=14976
01/01/01 AS MMDDYY8.
01-01-01 AS YYMMDD8.

January 1, 2001 AS WORDDATE25.
Monday, January 1, 2001 AS WEEKDATE29.
01JAN2001 AS DATE9.
01/01/2001 AS MMDDYY10.
2001-01-01 AS YYMMDD10.

(OTHER DATES DELETED FOR SPACE)

YEARDATE=24107
01/01/26 AS MMDDYY8.
26-01-01 AS YYMMDD8.

January 1, 2026 AS WORDDATE25.
Thursday, January 1, 2026 AS WEEKDATE29.
01JAN2026 AS DATE9.
01/01/2026 AS MMDDYY10.
2026-01-01 AS YYMMDD10.

11:19 Monday, August 12, 1996 8
0000000016, Year 2000 Test for SAS Formats (Using OS/2)

OBS YEARDATE J

1	01/01/2000	.
2	01/01/1900	.
3	02/29/1996	.
4	01/01/1900	.
5	03/14/1955	.
6	01/01/1901	.
7	01/01/1960	.
8	08/12/1996	.
9	02/29/2000	.
10	02/29/1996	.
11	08/12/1996	.
12	01/01/1997	1
13	01/01/1998	2

14	01/01/1999	3
15	01/01/2000	4
16	01/01/2001	5
17	01/01/2002	6
18	01/01/2003	7
19	01/01/2004	8
20	01/01/2005	9
21	01/01/2006	10
22	01/01/2007	11
23	01/01/2008	12
24	01/01/2009	13
25	01/01/2010	14
26	01/01/2011	15
27	01/01/2012	16
28	01/01/2013	17
29	01/01/2014	18
30	01/01/2015	19
31	01/01/2016	20
32	01/01/2017	21
33	01/01/2018	22
34	01/01/2019	23
35	01/01/2020	24
36	01/01/2021	25
37	01/01/2022	26
38	01/01/2023	27
39	01/01/2024	28
40	01/01/2025	29
41	01/01/2026	30
42	01/01/2026	31

N = 42