WIN100IN for Year 2000 Window Installations

Charles Edwin Shipp, Millennium Technologies Institute, Inc.

ABSTRACT

The procedure presented, WIN100IN, is JCL and SAS™ code that can automatically install either Fixed or Sliding 100-year Windows. It is particularly useful when applied to (pds) partitioned data sets that contain your SAS program source-code libraries. It can also be modified for TSO or CICS to interactively and automatically install windows in SAS programs stored in sequential data sets. For Fixed Windows, the user specifies the pivot year, for the starting year of the 100-year window, and for Sliding Windows, the user specifies the years back to position the window; it moves each year. A warning message is added, to begin appearing when it is time to archive data that will otherwise move out of the 100-year period.

WIN100IN reads a partitioned data set that contains SAS programs, one program per member, and creates a new pds with a fixed or sliding 100-year window installed in each program.

INTRODUCTION

The Year 2000 computer problem comes from needing to distinguish two-digit year representation in the 2000's from two-digit year representation in the 1900's. Since this represents common usage, and since there are now solutions to continue using two-digit representation in applications, the use of 100-year windows is well advised. The best Year 2000 knowledge and advice includes 100-year windows for most applications.

That is to say, you can combine the good original reasons for using two-digit year representation with the techniques of the new science of Year 2000. Appropriately use four-digit year representation when year spans are over say 75 years. Do the right thing because it is right, not because it is necessary and/or expedient. Although a lot of expense can be saved in time and computer disk resources by applying the techniques of Year 2000 100-year windows, both fixed and sliding, you want to apply what is appropriate.

Now that it is understood that the majority of SAS application programs can use the YEARCUTOFF feature and avoid retooling interfaces, databases, and programs, the programmer can easily edit all the programs and insert YEARCUTOFF at the top of each program. WIN100IN is able to automate this editing and makes it easier, especially when the sliding window is chosen. It is important to install a timed warning message since programmers / decision makers need to move the window in 5 to 95 years for fixed windows, and archive old data when using sliding windows.

When source libraries are stored in partitioned data sets, then WIN100IN can accomplish the installations with one setup and execution for each pds (partitioned data set.) It is very important in Year 2000 work to understand what you are doing. Fortunately, it is simple in concept and one only needs to read a few good papers to understand that most stand-alone SAS application programs or contained systems can continue to run well using the YEARCUTOFF 100-year window. (Assuming correct program/data usage.)

FIXED WINDOW

The SAS feature, YEARCUTOFF, available in all recent releases of SAS software®, allows a program to correctly interpret two-digit year representation in your specified 100-year time period.
Here is an example: Choose 1950 as the pivot (starting) year. The reason for this is that if your program now or in the future interfaces with other windowing programs, then they should all use the same window unless there is a good reason, such as: "you are certain the exchanged data remains in both windows."

Programs do best when they emulate common usage, and at the same time protect users from error. The 100-year window, invented by Year 2000 professionals, is an example of this.

The idea of the 100-year window is to specify the beginning year of the window, and then any two-digit representation, including '00', maps into a four-digit year value in that time period. It is that simple.

Look at it this way:

Another name for the beginning year, 1950, is the pivot year, and you can see why. Also, if you think about it, you can see why SAS Institute calls this feature YEARCUTOFF; (looking back in time to a certain year to include.)

The SAS statement to implement the illustrated 100-year window is

```
OPTIONS YEARCUTOFF=1950;
```

This statement can be added at the top of your SAS program, or you can modify an existing option statement. When this SAS feature is used, all proper date usage using two-digit year values will be properly interpreted. (Non-standard usage remains problematic. For this, refer to the other Year 2000 papers herein.)

With one simple change you solve the Y2K problem in a majority of SAS application programs/systems.

Warning: your data needs to stay in the window. Since the window is so big this is no problem for the majority of applications. Still, it is commendable to put a warning message that will begin to print, and will need to be addressed, in 95 years for sliding windows, and 45 years for the standard 1950 Fixed Window.

**CONSIDER YOUR DATA**

Some problems come into play with interfacing programs, or even with the program's own data, when data gets outside of the operative window. Most good SAS application programs have already considered when four-digit year representation should be used, and it is only mixing old and new data and going across the discontinuity of '98, '99, '00, '01 that two-digit years need to be analyzed. Also, good sorting programs, such as SYNCSORT and the IBM sort program, currently have 100-year window options to sort in proper order when you specify where the year values lie and what pivot year you want to use.
For data that spans many years, and is now or will soon be, in both centuries, use four-digit year representation. For data capture that is specific to the present, such as date of purchase, continue to use two-digit year values and apply YEARCUTOFF to those specific fields. The way that this works is to specify the SAS option at the top of the program, and then two-digit values are mapped into the specified 100-year window, and four-digit year values (of course) are already uniquely specified.

Your data, (its content and usage), determines your best Year 2000 solution choices. The most popular solutions are (1) 100-year windows, and (2) full-date expansion.

The disadvantage of full-date expansion is that it gets away from common usage, can be too expensive to convert, and may force other interfacing systems to change also.

The advantage of the window solution is that it is so easy and elegant, and maintains common usage of mm/dd/yy, which is preferred for human interfaces (screens and reports) with mm/dd/ccyy for database and electronic interchange.

When considering your data, remember that it is "field by field", not "program by program", that decisions are made. As an example consider a single program that has birth dates from 1890 through 2002 (very soon now). Obviously the birth date field must be four digits year values since the range exceeds any 100-year window. On the other hand, another variable could be for point-of-transaction dates and represent current dates in the current year, (or some other narrow time span.) These can be mm/dd/yy.

So the nature of the date fields in the data (each considered singly and separately) determines if you can use the 100-year window on that particular field or if you need to have four-digit representation for some specific fields. Remember that you can have fields of both nature in the same program, and YEARCUTOFF will automatically handle both of them as appropriate. Just remember to stick with standard SAS date functions—this is very important.

**SLIDING WINDOW**

The sliding window uses the same concept and techniques as the fixed window, plus one additional feature: the pivot year is calculated when the program begins execution. This way the 100-year window is positioned consistently over the current date by observing the system clock when execution begins. It is the programmer’s choice how to position the window during the setup calculation. For example, a good choice would be one year ahead, the current year, plus 98 years back; or five years ahead, the current year, plus 94 years back.

Suppose you want maximum coverage back, and minimum future coverage—to not waste all of your window years in the future with “only” 50 years back when you could have 99 years back. You may also want to set up a perpetual window that moves automatically each year. If so then choosing a Sliding Window may be best for you. Each time the program runs, it calculates once where to set the window and then issues the YEARSBACK or YEARCUTOFF option statement.

```
<table>
<thead>
<tr>
<th>1900</th>
<th>2000</th>
<th>2100</th>
</tr>
</thead>
<tbody>
<tr>
<td>'02</td>
<td>&quot;100-year window&quot;</td>
<td>'01  (for year 2001)</td>
</tr>
<tr>
<td>'03</td>
<td>&quot;100-year window&quot;</td>
<td>'02  (for year 2002)</td>
</tr>
<tr>
<td>'04</td>
<td>&quot;100-year window&quot;</td>
<td>'03  (for year 2003)</td>
</tr>
</tbody>
</table>
```
The only restriction is to not run old data (more than say 90 years old) into the program. That would be bad, but you should be able to segregate data appropriately. If this is a potential situation, you should be using four-digit year values anyway (and probably are for variables of this nature.) To assure that this is guarded against, a timed message is written into the SAS program (with the window) to begin appearing in 45 to 95 years for new programmers, just in case they need a reminder to archive that old, old data.

AUTOMATED SOLUTION — HOW IT WORKS

The program procedure uses SAS Proc SOURCE to write the original pds into a temporary sequential data set. The procedure then parses lines of each member for the first DATA or PROC statement, and puts the fixed or sliding window before it, with an appropriately timed warning statement.

The final part of the procedure uses the IBM utility, IEBUPDTE, to read the second temporary sequential data set and write the final pds source library.

Hence, the procedure herein is set up to convert a partitioned data set of non-compliant SAS programs to be Y2K compliant by inserting either a fixed or sliding 100-year window into each SAS program.

Before beginning, decide between fixed or sliding window, the pivot year or years back, and the names of temporary file space, pre-allocated or dynamically allocated. After running the procedure and testing, the original pds can be deleted or renamed for temporary holding (or 1. rename; 2. allocate; 3. run WIN100IN.)

SUMMARY

A majority of SAS application program systems can be made Year 2000 compliant (meaning they work as intended beyond 1999) by using the SAS YEARCUTOFF option. With one statement at the top of each SAS application program, the SAS date functions in the program understand that if you have four-digit year values then it is clear what year is represented, and if you have two-digit year values, then the four-digit year is the one that is mapped via the 100-year window specified. What could be easier? It is not only good, but also best, for most systems. It is better that four-digit year representation for the original reasons that two-digit year values are used in the first place: “common usage” and economies.

Before proceeding, understand the nature of the problem and your options for solution; and the nature of your data and system purpose and scope. This procedure allows you to convert large source libraries of SAS programs easily and efficiently. It has the advantage of putting in timed warning messages, and nobody does that, (but most should, i.e., “Do you want the same blame in the next century?”)

Sliding windows are easily put in and have the advantage of not becoming obsolete. Use careful data management though, as old data can get out of the window. Use great care in data exchange and give first consideration to passing four-digit year values. (Bridge programs are useful here.)

CONCLUSIONS

Full-date expansion (converting to four-digit years) is a good and excellent idea for some systems that are not too involved. For other systems it is not an option and would be only at great expense. Some involved systems are not worth converting, but a simple window adjustment makes them viable again.

In most legacy SAS systems, putting in a fixed 100-year window is the best solution, and in others the sliding window is best. If they both seem equally good to you then either one will serve you well. The WIN100IN procedure makes an easy task easier and better. (You need to manage 50-year data blocks.)
APPENDIX A: FUNCTIONAL (OVERVIEW) DIAGRAM

(1) Input pds

index
member_1
member_2
member_3
...
...
...
...
member_N

(2) WIN100IN

- JCL
- SASpgm
- IBM Util

(3) Output pds

index
member_1
member_2
member_3
...
...
...
...
member_N

APPENDIX B: DETAIL DIAGRAM

WIN100IN

SAS Proc SOURCE

Sequential File 1
(temporary)

Sequential File 2
(temporary)

IBM Utility IEBUPDTE

SAS logic to put in one line & message!

APPENDIX C: ONLINE SOURCE FOR WIN100IN

The code for WIN100IN will be placed in the SAS Exchange Library, after it finds a new home, (on SAS Web Page?) or contact the author: <CharlieShipp@CompuServe.com> and I will send you the code via Email.