"Had a Strange DATE" – problem solved. Now as SAS gets older will ANYDT do?
Venky Chakravarthy PhD, BioPharma Data Services, Ann Arbor, MI

ABSTRACT
This is a sequel to a 2002 paper that was written before the ANYDTxxx INFORMATS were introduced. This paper draws on the salient points from that paper and examines the pros and cons of the ANYDTDTE INFORMAT. While recognizing the uses of this new family, caution is advised in using them as there are some pitfalls for the unwary.

INTRODUCTION
This is a sequel to an earlier paper by the author “Have a Strange DATE? Create Your own INFORMAT to deal with her” (Chakravarthy, 2002). In keeping with the tradition of the lighter tone of the earlier paper this paper attempts to do the same. Hopefully, the sequel lives up to the reputation of the predecessor.

Despite the frivolous title of this paper and some of the headers within, this is a serious discussion. You, as a SAS® user, often deal with dates err… date values. Sometimes you are reporting a date value in a form that can be supplied with an available SAS FORMAT and at other times you are reading in a date value with a SAS date INFORMAT. You are also (often) required to report a date value in a form that cannot be produced using a SAS supplied date FORMAT. Likewise, you are also trying to read in a date value for which there is no SAS date INFORMAT. In short, the form a date can take a value that is limited only by the imagination of your customer. You have to get creative in dealing values that a FORMAT/INFORMAT cannot readily handle. Consider one poster on the SAS-L forum who asked, "I was wondering whether anyone had a solution to read the following data line using a SAS INFORMAT: December 20, 1999". While the WORDDATE format can OUTPUT such a value, there is no corresponding SAS date INFORMAT (see update for Version 9 later in the discussion) to INPUT such a value. How do you then input this date value? Prior to version 9, we did not have an INFORMAT to handle this. We will see how an accidental feature has been enabled in Version 9 that allows you to read a WORDDATE value with one of the INFORMATS.

This paper starts with a discussion of dates that are simple to deal with. Progressively, it discusses ways to write and read date values that are beyond the scope of ready handling with existing FORMATS and INFORMATS. The paper also discusses how to create INFORMATS to input such date values. Finally, it compares some of the merits and demerits of creating INFORMATS with another method of inputting such dates.

Parts of this paper draw heavily from its predecessor (Chakravarthy, 2002). So portions of the paper are deliberately kept brief in order to avoid repeating concepts discussed before. A basic understanding of SAS date FORMATS and INFORMATS is assumed. Also assumed is an awareness of the SAS date FUNCTIONS and some familiarity with PROC FORMAT. The target audience is both the beginner and the advanced user.

THE TYPE OF DATES DISCUSSED
We shall begin by briefly examining the “types” of dates seen in SAS. Broadly we can classify them as follows.

1. “Good” DATE – Can be safely brought home - Has both FORMAT and INFORMAT
2. “Not so good” DATE – Hesitate a little - Has FORMAT but NO INFORMAT
3. “Strange” DATE – Stop! - Has No FORMAT and No INFORMAT
4. “Any” DATE – Not sure yet - Appears to settle for ANYDT

The fourth point is intriguing and the only one not covered by its predecessor. This will be covered in detail in this paper. Points 1 to 3 will be briefly discussed to lead naturally to the topic at hand. The reader is encouraged to read the earlier paper. Let us go on with the brief review of the first 3 and elaborate more on ANYDT.

THE “GOOD” DATES
Let us start expanding on the above topics and start with a “Good” DATE. An example is a date in the form of DDMONYYYY – 22MAY2010. The DATE9 FORMAT outputs date values in this structure. If an external file has such values, it can be read in using the DATE9 INFORMAT.
THE “NOT SO GOOD” DATES
This type of DATE has either a FORMAT or an INFORMAT but not both. An example of such a date value is WORDDATE. You can run this one line code and write to the log today’s date in WORDDATE format.

```sas
%put %sysfunc(putn(%sysfunc(today()),worddate.)) ;
```

If you are an advanced SAS user, you can refute this absolute statement that this date has no INFORMAT and you are right. We will get to that later but for our immediate purposes this is a reasonable example. If we try using a WORDDATE as an INFORMAT this is the message received in the log.

ERROR 48-59: The informat WORDDATE was not found or could not be loaded.

There are several ways to input these values which were covered in the predecessor paper (Chakravarthy, 2002). The various solutions will not be repeated here. However, one solution to this type of problem is to create a user INFORMAT as follows.

```sas
data cntlin ;
  retain fmtname "baddate" type "I" ;
  do label = "01Jan2000"d to "31dec2020"d ;
    start = trim(put(label,worddate.-L)) ;
    output ;
  end ;
run ;
```

```sas
proc format cntlin = cntlin ;
run ;
data _null_ ;
  input d baddate. ;
  put "PharmaSUG 2010 Start Date is " d : worddate. ;
cards ;
May 22, 2010
run ;
```

Note that we are limited by the range of the dates that we store in the INFORMAT. However, a user can easily accommodate a few centuries of DATE values without taxing the computer. That should be enough for any real life encounters with “Not So Good” DATES.

THE “STRANGE” DATES
The “Strange” DATES are those that require extensive hand holding. You may have to summon enough courage to deliver some strong DIRECTIVES for these “Strange” DATES to behave. Intrigued? You will find more details in the earlier paper cited many times before. Here is a hint: creating custom date formats were simplified in SAS from version 8 using the DATE DIRECTIVES.

PERFORMANCE CONSIDERATIONS
The performance of different approaches to reading DATE values were empirically measured previously and are reproduced below for the reader’s benefit.

<table>
<thead>
<tr>
<th>INFORMAT APPROACH</th>
<th>OTHER APPROACH</th>
<th>DIFFERENCE</th>
<th>RATIO</th>
</tr>
</thead>
<tbody>
<tr>
<td>38.16</td>
<td>82.13</td>
<td>-43.97</td>
<td>0.46</td>
</tr>
<tr>
<td>39.92</td>
<td>84.42</td>
<td>-44.50</td>
<td>0.47</td>
</tr>
<tr>
<td>37.91</td>
<td>81.75</td>
<td>-43.84</td>
<td>0.46</td>
</tr>
<tr>
<td>38.14</td>
<td>81.95</td>
<td>-43.81</td>
<td>0.47</td>
</tr>
<tr>
<td>37.95</td>
<td>81.69</td>
<td>-43.73</td>
<td>0.46</td>
</tr>
</tbody>
</table>
THE “ANY” DATE APPROACH

SAS has aged now with many good youthful years left behind. In its middle age years settling for “ANYDT” seems to be a fine approach. Or is it? Let us examine these set of INFORMATS first introduced with Version 9. We have three different types in the ANYDTxx family.

1. ANYDTDTE
2. ANYDTDTM
3. ANYDTTME

We can see that only the last three letters differ. The first is used to read a variety of values and store as DATE values. The second reads similar set of values and stores as DATETIME values and the third as TIME values. To provide continuity to the previous paper and the current discussion, we will discuss only the ANYDTDTE INFORMAT. It can be easily extended to the other two by the reader.

The ANYDTDTE is simply a way to subtly convey to the user that they can read in ANY SAS DATE value. That is great – Now I can stop reading this paper. You can stop reading this paper. All your “DATE” problems have been solved. Hold that thought. Perhaps, not and you may still want to finish reading this paper and here is why. First and foremost the SAS documentation does not claim that it can read ANY date value but only ANY of the following 9 types of date values:

DATE, DATETIME, DDMMYY, JULIAN, MMDDYY, MONYY, TIME, YYMMDD, or YYQ

A few more types of date values have been added in Version 9.2 e.g. MDYAMPM. The point here is that it is an impossible task to attempt to cover all types of date values. What ANYDTDTE does is rather simple. It uses an existing INFORMAT or an internal INFORMAT under the hood that is not documented in the public domain. Let us run the example that is in the documentation.

```sas
data _null_;
  input checkdate : anydtdte.;
  put checkdate=:date9.;
cards;
14JAN1921
14JAN1921 12:24:32.8
14011921
1921014
01141921
JAN1921
12:24:32.8
19210114
21Q1
run;
```

Here is the log for the values.

checkdate=14JAN1921
checkdate=14JAN1921
checkdate=14JAN1921
checkdate=14JAN1921
checkdate=14JAN1921
checkdate=01JAN1921
checkdate=01JAN1921
checkdate=01JAN1960-------------------------------------------{1}
checkdate=14JAN1921
checkdate=01JAN1921

We notice that it does a nice job of capturing a variety of dates just as advertised. Here are a few things to bear in mind. If a time value is read, the default date assigned is 01JAN1960. So the value is read but it does not have much use.
DATESTYLE SYSTEM OPTION

If a value of 11/05/2009 is encountered, is that November 5, 2009 or May 11, 2009. This is simple. There is an option called DATESTYLE which is set to LOCALE by default. If you are in England, in this global economy with massive amount of data transfers something can easily slip up especially if the rendering software is not capable of recognizing this. Let us briefly run some code with and without the DATESTYLE option that has the following range of values: DATESTYLE = MDY | MYD | YMD | YDM | DMY | DYM | LOCALE

```sas
data _null_;  
  input text : anydtdte. ;  
  put text = date9. ;  
  cards ;  
11/05/2009  
run ;
```

The log values is: text=05NOV2009

Now let us do the same this time to our customer in England.

```sas
options datestyle = DMY ;
 
data _null_;  
  input text : anydtdte. ;  
  put text = date9. ;  
  cards ;  
11/05/2009  
run ;
```

Now the log has this value: text=11MAY2009

This is great and we can customize the reading of values based on the locale. SAS will read 01/02/03 as February 1, 2003 in England, Jan 2, 2003 in the US and in Japan as Feb 3, 2001. However, it has to be emphasized that the DATESTYLE cannot be changed mid stream. In other words if you receive a batch of data and input it the DATESTYLE is set for the entire data. This would cause a great deal of concern. It is worse when the data has all three types. Taking a less cynical view, this would be very unlikely since any rendering software will have some consistency, at least, within the same output.

WORDDATE AUTOMAGICALLY READ WITH ANYDTDTE

We had earlier seen how WORDDATE is an invalid INFORMAT. Somehow this INFORMAT is available under the hood for ANYDTDTE. That is great news. The Online documentation for Version 9.2 has been updated to reflect this under the ANYDTDTE section but still does not explicitly state that the form is WORDDATE. Here is an example of how it works:

```sas
data _null_;  
  infile datalines truncover;  
  input x anydtdte20.;  
  format x date9.;  
  put x = ;  

datalines;  
May 22, 2010  
run;
```

Log: x=22MAY2010

CAUTIONARY NOTES WITH ANYDTDTE

There are also a few bugs associated with ANYDTDTE. It reads in some impossible values.

```sas
225  data _null_;  
226    dob=input('110009',anydtdte6.);  
227    put dob mmddyy10.;  
228  run;
```

03/12/2261
A second bug is the inability to read certain date values. Not only that but it does not issue a note.

```sas
data _null_;  
  input date anydtdte8.;  
  format date mmdyyyy10.;  
  put date=;  
  datalines;  
15122005  
16122005  
17122005  
21122005  
22122005  
run;
```

The following is issued in the log:

```
date=12/15/2005  
date=.  
date=.  
date=.  
date=12/22/2005
```

The good news is that these bugs are known issues and SAS has addressed some in Version 9.2. Other bugs are marked for fixing in a future release. So this should not be a concern in the future. However, at the time of writing this paper many large organizations have not adopted Version 9.2 in production so any use of ANYDTDTE would do well to heed this cautionary note.

CONCLUSION

We have seen “Good”, “Not So Good”, “Strange” and “Any” dates and concluded that DATES could be a problem if you are trying to read them. The introduction of ANYDTDTE solved some problems while introducing new ones. The user is advised to be cautious when reading in dates rendered from non-standard packages. There are potential pitfalls with using any solution, but one must be careful with ANYDTDTE. Familiarize yourself with your data – the old adage is still good advice.

REFERENCES


CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

- Name: Venky Chakravarthy  
- Enterprise: BioPharma Data Services  
- E-mail: Venky@biopharmadataservices.com  
- Web: http://www.biopharmadataservices.com/

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.